Statistical inverse problems for population processes

Sollie, Birgit

2021

document version
Publisher's PDF, also known as Version of record

Statistical inverse problems for population processes

Birgit Sollie

Cover design by Ronald Brands @MerkMeester

VRIJE UNIVERSITEIT

Statistical inverse problems for population processes

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor
aan de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. V. Subramaniam,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de Faculteit der Bètawetenschappen
op maandag 7 juni 2021 om 9.45 uur
in de aula van de universiteit,
De Boelelaan 1105

door

Birgit Sollie

geboren te Lelystad

promotoren: prof.dr. M.C.M. de Gunst
prof.dr. M.R.H. Mandjes

Acknowledgement

'Give thanks to the Lord, for he is good;
His love endures forever.' Psalm 107:1.

With great joy I take this opportunity to thank everyone who has supported me during my PhD and helped me to complete this thesis successfully. I could not have done it without all these people.

I would like to express my sincere gratitude to my supervisors, Mathisca de Gunst and Michel Mandjes, who dared to embark on this adventure with me. Thank you for your encouragements and for your confidence in me. You were always convinced of a positive outcome, even in more difficult times. I thank Mathisca for being a role model as a female professor, and for sharing her knowledge with me. It was a great pleasure to work with you and to learn from you, especially regarding our joint interest in statistics for biological processes. I thank Michel for his devotion and enthusiasm for the projects. You were always full of ideas and our meetings consistently gave me new energy. I have learned a lot from you. I also thank Bartek Knapik, my daily supervisor in the first two years of my PhD. You helped me to get started as a researcher and was always there to help me with my questions. It was a pleasure to collaborate with you on the first paper.

I thank Geurt Jongbloed, Harry van Zanten, Sandjai Bhulai, Stéphanie van der Pas and Sophie Hautphenne, for being willing to take place in my doctorate committee. Thank you for your time to read my thesis and for your presence at my defense. Sophie, I would like to thank you for our pleasant collaboration. I have enjoyed working with you and learning from you. Thank you for all your constructive comments, which improved my work considerably.

My time as a PhD student would never have been the same without all the friendly colleagues and office mates I had over the years. Thank you for all the lunch and coffee breaks, the good conversations and the fun we had, especially during conference visits and meetings.

I am grateful for being part of the NETWORKS program. I have experienced it as a pleasant environment surrounded by so many kind fellow researchers. The training weeks were highlights during my PhD, where I had the opportunity to learn about so many interesting topics, but where we also had a lot of fun together at the social activities and when playing boardgames. I want to thank everyone within NETWORKS for these amazing times.

My expression of gratitude is not complete without thanking all the people who support me at home. I thank all my friends, with in particular Angela, Anne, Maartje, Milou, and Nadieh. Thank you for all your support and for all the fun we have together. I am incredibly grateful for our friendship.

I thank my family for all their support. I am especially grateful to my beloved parents, Anja and Freek. Thank you for giving me a loving family to grow up in and for giving me the opportunity to go to university and study Mathematics. I am where I am because of you. I cannot thank you enough for everything you have done for me.

Last, but not least, I thank Tibor, my soulmate, who I may now call my husband. You have been so valuable for me in these past years. Your love and encouragements were indispensable. I am blessed to have you in my life. Every day with you is a lovely day and predicts a happy future together.

I believe that my competence comes from God, my Saviour, Father and Friend. I can do all things through Him who gives me strength.

Birgit, *March 2021*

Contents

1. Introduction

Since the beginning of 2020, the world is struck by the coronavirus, officially named COVID-19. At this moment, the virus is still spreading among people all over the world. Figure 1.1 shows the number of reported COVID patients per day in the Netherlands from February 27, 2020 to January 27, 2021. Importantly, the virus is only detected on people that are tested. Therefore, the actual number of COVID patients is presumable higher than the numbers shown in Figure 1.1. Natural questions arise; 'how fast does the virus spread?', 'what is the infection rate?' and 'which factors affect this rate and how?'. Note that the amount of untested, and hence undetected, COVID patients, who for example do not have any symptoms, does affect the spread of the virus. This thesis gives insight in how to tackle the kind of questions mentioned above, and shows in particular how to deal with unobserved factors that affect the rate(s) of interest.

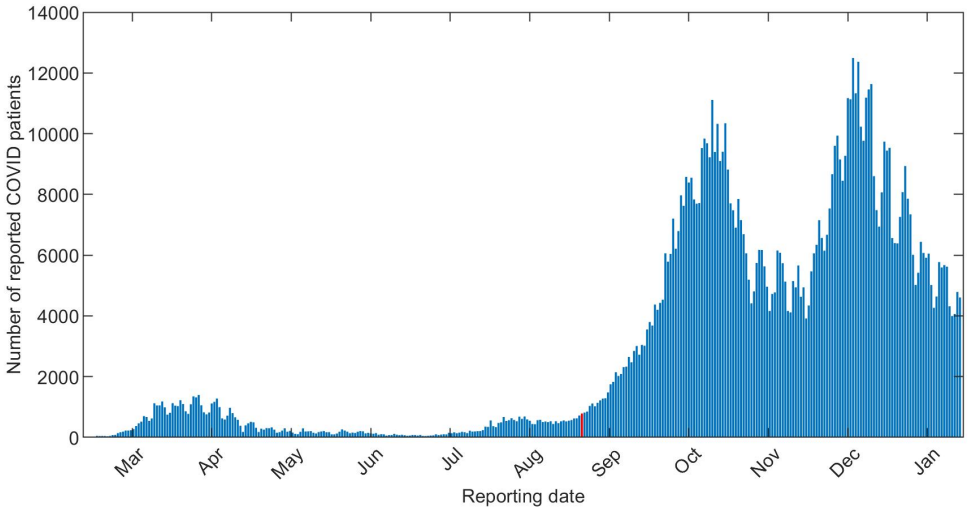

Figure 1.1: Number of reported COVID patients per day in the Netherlands from February 27, 2020 to January 27, 2021. The author's own reporting date, September 4, is shown in red. (*Source: RIVM*)

Research objective

Throughout this thesis we consider a class of models known as *population processes*. Population processes are stochastic processes that record the dynamics of the number of individuals in a population, and have many different applications in a broad range of areas such as biology, economics and operations research. They are, for example, suitable for modeling the spread of infectious diseases when the individuals are considered to be the infected people in a population. However, it is important to emphasize that the individuals in population processes do not necessarily have to be people. For instance, one could also think of a model for a population of animals, a model for molecules in a cell, or even a model for the number of visits to a website.

Population processes are stochastic processes, hence transitions in the population size do not occur at fixed times, but according to some probabilistic mechanism. Furthermore, population processes are often modelled as Markov processes which means that the transition probabilities do not depend on the past, but on the current population size only. An important feature of population processes is that transitions correspond either to an increase or decrease in the population size. These two types of transitions are often referred to as *births* and *deaths*, or the analog terminology *arrivals* and *departures* is used, respectively. The *lifetime* of an individual is the time between its birth and its death. A specific class of population processes is the class of birth-death (BD) processes, where transitions can only increase or decrease the population by one at a time. A commonly used assumption for population processes is that there is no interference between individuals in a population, in the sense that the lifetimes of the individuals are independent. Under this independence assumption the resulting BD process can be seen as an infinite-server queue, a key model originating from queueing theory.

It is usually assumed that the lifetimes are exponentially distributed, and that the births follow a Poisson process. However, in many situations the dynamics of the population is affected by exogenous, often unobservable, factors, as we noticed for the COVID data. Think of temperature affecting the spread of a bacteria or weather conditions affecting the mobility of individuals. This results in a higher variability in some, or all, model parameters, which we want to include in the population process. We do this by adding an underlying stochastic process to the model, referred to as the *background process*, which affects the parameters of the population process. Together, the population process and the background process form a bivariate Markov process. For clarification we continue with an example.

Example 1.1. We consider a population process of which the births follow a Poisson process and the lifetimes of the individuals are independent and exponentially distributed. The background process is a continuous time Markov process with two states, of which the state determines the rate of the Poisson process. While the background process is in state 1, the parameter of the Poisson process is equal to λ_1. However, as soon as the background process jumps to state 2, the parameter of the Poisson process switches to λ_2, see Figure 1.2 for a schematic representation. This specific example is also known as a *Markov-modulated* process, in which the parameters switch between multiple distinct values at the jump times of a modulating background process. To illustrate how the background

process affects the population size, we include the results of a simulation of this example, see Figure 1.3. It shows the population size in the upper panel simultaneously with the state of the background process in the lower panel. There are fluctuations visible that are clearly the result of the changes in the state of the background process. Note that in this example only the births are affected by the background process. This thesis will consider models in which the deaths are affected by the background process as well.

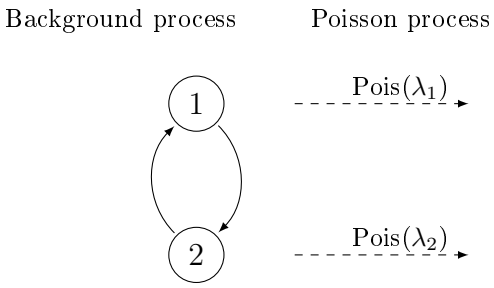

Figure 1.2: Schematic representation of a background process affecting the rate of a Poisson process.

We are interested in answering questions regarding the parameters of a bivariate model consisting of a population process together with an underlying background process. To this aim, we need reliable techniques to estimate the model parameters, including those related to the background process. The difficulty in finding these techniques is largely determined by what exactly can and cannot be observed. Throughout this thesis, we make the following three model assumptions, which make the statistical inference challenging:

- The background process cannot be observed. The challenge is to still infer the parameters of the background process, and how this process affects the population process.
- Only the population size is observed. The challenge is to infer the parameters from the birth- and death processes separately from observations of the *net effect* of these processes.
- The population process can be observed only at a finite number of deterministic points in time, which reflects the fact that in most practical situations it is infeasible to observe the process continuously in time. The challenge here lies in the fact that it is unknown what happened in between two consecutive observations.

Note that the combination of the second and the third assumption increases the degree of complexity substantially. For example, if the births and deaths are discretely observed separately, information is lacking on the exact times of the births and deaths in between two consecutive observations. However, if the population size is discretely observed only, additional information is lacking on the number of births and the number of deaths in between two consecutive observations. Combined with the first assumption, this results in complex inverse problems.

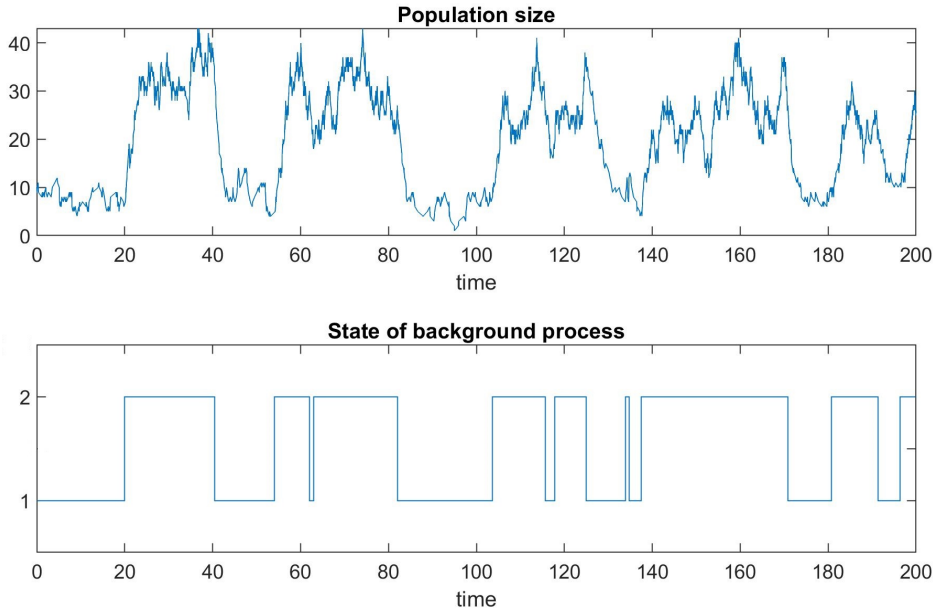

Figure 1.3: Upper panel: number of individuals in the population. Lower panel: state of the background process.

Statistical inference for population processes

Population processes affected by underlying background processes have been studied considerably over the years. However, inverse problems for these models are considered substantially less often. In the individual chapters we present extensive surveys on existing literature. Here we would like to provide and highlight a few important contributions.

With regard to statistical inference for population processes, we mention [18, 19, 24, 67, 70] for various parameter estimation procedures. In these papers, the population process is not affected by an unobserved background process, but the analysis is complicated by the fact that the population is observed at discrete times only, which means that the individual births and deaths are not observed directly. In [31, 32], statistical inference is performed for population processes that are affected by a Markovian background process. More specifically, these papers study parameter estimation problems for Markovian binary trees, based on continuous-time observations. For the case of discrete-time observations, parameter estimation results for the class of Markovian arrival processes are given in [50, 15]. The data in these last two papers directly concern the cumulative birth process, as the models do not include deaths, however here the analysis is complicated by both an unobserved background process and by discrete-time observations.

In the context of queueing theory, there is a body of work on statistical inference for population processes as well. As mentioned above, infinite-server queues can be viewed as a specific type of population processes. In the terminology of queueing theory, arrivals and departures are used as the analog for births and deaths, respectively. For infinite-server queues, the service times can be viewed as the lifetimes of the individuals. Infinite-server

queues have been studied extensively, and attention has been paid to infinite-server queues in a random environment. We refer to [47] for more background information on these kind of models. The steady-state behavior and moments of infinite-server queues in a random environment, with known parameters, are analyzed in [48]. For more specific analysis on Markov-modulated infinite-server queues, performed under particular scaling of the model parameters, we refer to [12, 13, 14]. The papers above do not perform any statistical inference, but give insight in important properties of the models. Parameter estimation procedures have been proposed for infinite-server queues that are not affected by unobserved background processes. We mention [55], where the estimation of the arrival rate is performed, based on the method of moments and the maximum likelihood procedure. Here it is assumed that all arrivals and departures are observed, without knowing which departure belongs to which arrival. In [11] a parametric and a non-parametric procedure are derived for estimating the arrival rate and the service time distribution based on continuous-time observations of the population size.

The complexity of inverse problems for population processes depends on both the complexity of the model and, as described before, the assumptions regarding the observations. When looking at the three model assumptions listed above, it stands out that in existing literature often one of the listed assumptions is included, sometimes even two. However, to the best of our knowledge there is no existing literature that includes all three of the listed model assumptions. The research in this thesis distinguishes itself by the fact that all three model assumptions are included in the inverse problems.

Thesis outline

This thesis covers a wide range of models which all have in common that they are population processes affected by an unobserved background process. The aspects in which the models differ, result in a need of specific inference techniques. In line with the different chapters of this thesis we present an overview of our contributions with emphasis on the differences in the models and the techniques.

The first class of models covered in this thesis is a certain class of Markov-modulated population processes. Example 1.1 above already introduced this model for the special case of a two-state-background process. **Chapter 2** considers Markov-modulated population processes of which the background process can have any finite number of states. A method is established to estimate all model parameters, including those related to the modulation, based on discrete-time observations of the population process. The EM algorithm is used to develop an algorithm for finding maximum likelihood estimates of the parameters. This algorithm iteratively maximizes the likelihood function and at the same time updates the parameter estimates.

The class of Markov-modulated population process considered in Chapter 2 can be extended to the more general class of continuous-time quasi birth-death processes. **Chapter 3** introduces this class of models in terms of a bivariate Markov process. The goal is to evaluate the likelihood function based on discrete-time observations of the population process. To achieve this, we need insight in the transient behaviour of the bivariate Markov process. It is shown how the time-dependent distribution of the Markov process

can be numerically approximated in an accurate and efficient way, using the so-called Erlangization technique. With this time-dependent distribution, the likelihood can be evaluated and numerically maximized to find maximum likelihood estimates.

Chapter 4 can be seen as a follow-up chapter to Chapter 3. It considers a class of processes, which we call the class of on/off-seq-L processes, inspired by a specific biological application, namely, the number of mRNA molecules in single living cells. An on/off-seq-L process can be seen as a birth-death process of which the births are regulated by an on/off mechanism and follow a sequential process consisting of multiple steps. The goal is to evaluate the likelihood function based on discrete-time observations of the population process, and to estimate all model parameters. This in turn is applied to real-life mRNA data. An important step in reaching this goal, is the realization that an on/off-seq-L process is a special case of the quasi birth-death process covered in Chapter 3. Hence, we can rely on the technique introduced in Chapter 3 to approximate the likelihood function for the on/off-seq-L process.

Where the first chapters consider one-dimensional population processes, **Chapter 5** covers multivariate population processes, in which the population lives on a multi-node (rather than single-node) network. Besides the births and deaths that can occur on each node, individuals can move along the edges between the nodes of the network. In addition, the multivariate population process is again considered under Markov-modulation. The goal is to estimate all model parameters, based on discrete-time observations of the multivariate population process. Note that the model parameters now include the birth- and death parameters corresponding to each node, the parameters related to the movement along the edges, and the parameters related to the modulation. To deal with the high complexity of this model, a discrete-time population process is considered, in contrast to the continuous-time models in the chapters before. The likelihood function is then accurately approximated by applying saddlepoint approximations, which is a technique highly suitable for the network structure of the model. The likelihood function can again be numerically maximized to find maximum likelihood estimates.

All Chapters 2–5 described above, contain an extensive simulation study to investigate the accuracy of the inference method. **Chapter 6** completes the thesis with concluding remarks on the differences between the models and the techniques and with a look-out on further research. Chapters 2–5 can be read independently of each other, but Chapter 4 refers back to the Erlangization technique introduced in Chapter 3. Chapters 2,3 and 5 of this thesis are based on the following papers, respectively.

(i) M. de Gunst, B. Knapik, M. Mandjes and B. Sollie. Parameter estimation for a discretely observed population process under Markov-modulation. *Computational Statistics & Data Analysis*, 140: 88–103, 2019.

(ii) M. Mandjes and B. Sollie. A numerical approach for evaluating the time-dependent distribution of a quasi birth-death process. *Under revision.*

(iii) M. de Gunst, S. Hautphenne, M. Mandjes and B. Sollie. Parameter estimation for multivariate population processes: a saddlepoint approach. *Stochastic Models*, 37: 168–196, 2021.

2. Markov-modulated population processes

A Markov-modulated independent sojourn process is a population process in which individuals arrive according to a Poisson process with Markov-modulated arrival rate, and leave the system after an exponentially distributed time. A procedure is developed to estimate the parameters of such a system, including those related to the modulation. It is assumed that the number of individuals in the system is observed at equidistant time points only, whereas the modulating Markov chain cannot be observed at all. An algorithm is set up for finding maximum likelihood estimates, based on the EM algorithm and containing a forward–backward procedure for computing the conditional expectations. To illustrate the performance of the algorithm the results of an extensive simulation study are presented.

2.1 Introduction

Population processes have been studied extensively, owing to their applicability in a broad range of areas such as biology, medicine, economics and operations research. A specific type of population process is one in which individuals arrive in a system, but once in the system they do not interfere with each other, and their sojourn times—that is, the times the individuals spend in the system—are independent. In queueing theory, this is conveniently called an *infinite-server queue*. A commonly imposed assumption is that of Poisson arrivals, but in many real life systems the arrival process is substantially more variable, for example alternating between busy and quiet periods. In such situations the *Markov-modulated Poisson process* (MMPP) is a more suitable alternative. The MMPP is a doubly stochastic Poisson process of which the rate is determined by a finite, continuous-time Markov chain, also referred to as the background process, such that the rate switches between distinct values at the jump times of the modulating Markov chain.

In this chapter we consider a population process with independent sojourn times—or of the infinite server type—fed by an MMPP arrival process, and will refer to it as a Markov-modulated independent sojourn (MMIS) process. This class of models can be seen as birth-death processes under modulation, and has applications across various disciplines, see for example, [5, 48, 65]. We note that in queueing theory this process is also known as the $M/M/\infty$ queue in a random environment [48]. For more background information on this kind of stochastic models, see for example [47, Ch. 3 and 6]. We also

remark that the cumulative arrival process is a counting process.

We are interested in estimating the unknown parameters of this system—including those related to the modulation. We assume that the population size can be observed, but the modulating Markov chain cannot. More specifically, the number of individuals present is recorded only at equidistant points in time, which reflects the fact that in most practical situations it is infeasible to observe the process continuously in time. We further assume that the times the individuals spend in the system are independent and exponentially distributed with a constant (that is, non-modulated) rate parameter. Since in many applications the departure rate can be controlled or is otherwise known, we develop our estimation procedure for the case that this rate parameter is known. We then show that under a natural additional assumption, the departure rate parameter can be estimated together with the parameters of the arrival process in a similar way.

An analysis of the MMIS process in terms of generating functions and moments can be found in [48], while asymptotic properties have been investigated in [4, 14]; see also the recent paper [52]. In the setting of queueing systems several parameter estimation procedures have been proposed for the analysis of infinite-server queues with non-modulated Poisson arrivals. We mention the method of moments and the maximum likelihood procedure for estimating the arrival rate developed in [55]. Here it is assumed that all arrivals and departures are observed, without knowing which departure belongs to which arrival. Another relevant reference is [11], in which a parametric and a non-parametric procedure are derived for estimating the arrival rate based on continuous-time observations of the population size, as well as a non-parametric estimation method based on the idle and busy periods.

Outside the setting of queueing systems, substantial attention has been paid to parameter estimation for counting processes which are affected by a hidden background process. Because of the hidden background process, missing data are intrinsic to estimation in this context, and the EM algorithm [25] plays an essential role. The first class of models with a hidden background process for which parameter estimation was considered, is the class of discrete-time hidden Markov models. The Baum–Welch algorithm [56, 69], which is used for parameter estimation in hidden Markov models, is essentially an EM algorithm. Later on, various continuous-time processes have been considered as well. It has been shown how to apply the EM algorithm to the class of phase-type distributions [8], the class of MMPPs [63, 61] and, more generally, to the class of Markov-modulated Markov processes [27]. Rydén's EM algorithm for MMPPs can also be used for parameter estimation in Markovian arrival processes. An improved algorithm for this case has been proposed in [49]. In this body of literature estimation is performed based on observations of the counting process, the cumulative arrival process in the MMIS system. This is a marked difference with our setting, in which we wish to learn the system's input parameters from data on the population size. A specific case of parameter estimation for a Markovian arrival process from population size data is presented in [31], where Markovian binary trees are considered. This is, again, a setting very different from ours.

To the best of our knowledge, parameter estimation for MMIS processes under the particular assumptions on the data that we formulated above, has not been studied so far. In all papers listed above it is assumed that the counting process can be observed

continuously in time, and as a result, missing data only arise due to the hidden nature of the background process. Parameter estimation results for arrival processes based on discretely observed data are known only for the class of Markovian arrival processes [50, 15], for which naturally the EM algorithm was used as well. However, in these papers the data directly concern the cumulative arrival process as their models do not include departures, whereas in our setting we only indirectly observe the *effect* of the arrivals, namely through the population size.

In our context the parameter estimation is seriously complicated by (i) the fact that the modulating Markov chain of the Markov-modulated arrival process is not observed, and (ii) that the population size is not observed continuously in time. Issue (i) entails that it is not known when the arrival rate changes value, and issue (ii) that it is not possible to deduce from the observations the number of arrivals or the number of departures between two consecutive observations. To deal with these complications we treat the modulating continuous-time Markov chain and the number of arrivals between two consecutive observations as missing data, and, making use of the EM algorithm, develop an explicit algorithm to find maximum likelihood estimates of the parameters. Our approach borrows some elements of the one proposed by Okamura et al. [50], but the adaptation of their estimation algorithm to our setting is not straightforward. Although we end up with the same type of parameter updates as the ones they employ, the computations of these updates require major adjustments to the steps of their algorithm. In particular, we use a different method to obtain the required transition probabilities and we redefine the forward and backward vectors. The conditional expectations in the parameter updates can be expressed as integrals containing these vectors in a similar way as in [50], but the computations of these integrals now demand solving a significantly more involved system of differential equations.

The remainder of this chapter is organized as follows. In Section 2.2 we define our statistical model, the MMIS process, and we state the estimation problem. In Section 2.3 we derive the estimation algorithm. We investigate the accuracy of the proposed estimation method by a simulation study in Section 2.4. Section 2.5 discusses an extension in which the departure rate is not known and estimated as well. The chapter concludes with a discussion in Section 2.6.

2.2 Model and estimation

We consider an MMIS process where the arrivals follow an MMPP driven by a modulating continuous-time Markov chain of which each state corresponds to a different arrival rate value. In this section the mathematical formulation of this model and the corresponding estimation problem are presented.

2.2.1 Markov-modulated independent sojourn process

The modulating continuous-time Markov chain with state space $\{1,\ldots,d\}$, $d \geq 2$, that defines the state of the arrival process at time t, will be denoted by $\{X_t\}_{t\geq 0}$. Its transition rate matrix is given by $Q = (q_{ij})_{i,j=1}^{d}$ and its initial state distribution at $t = 0$ by $\pi = (\pi_1,\ldots,\pi_d)^\top$. We define $q_i = -q_{ii} = \sum_{j\neq i} q_{ij}$ as the total rate at which the Markov chain jumps out of state i. The MMPP models the cumulative arrival process $\{A_t\}_{t\geq 0}$ with corresponding time-inhomogeneous arrival rate $\lambda(t)$. This rate stochastically alternates between d different rates $\lambda_1,\ldots,\lambda_d$ in such a way that $\lambda(t) = \lambda_i$ if $X_t = i$, for $i = 1,\ldots,d$. We assume that the sojourn times are independent and identically exponentially distributed with rate $\mu > 0$.

Let $\{M_t\}_{t\geq 0}$ be the population size at time t. Then $\{M_t, X_t\}_{t\geq 0}$ is a joint Markov process with corresponding transition probabilities

$$p_{ij}(m,m';t) = \mathbb{P}(M_t = m', X_t = j \mid M_0 = m, X_0 = i), \tag{2.1}$$

for all $t \geq 0$ and $m, m' \geq 0$. For each combination of m and m', we define for $t > 0$, the $d \times d$ transition matrix $P_t(m,m')$ containing these transition probabilities by

$$[P_t(m,m')]_{ij} = p_{ij}(m,m';t). \tag{2.2}$$

We assume that the process $\{M_t\}$ is observed at $n+1$ equidistant time points $t_k = k\Delta$ for some fixed $\Delta > 0$, $0 \leq k \leq n$, and denote the corresponding observations by $m_0,\ldots,m_n$. These $n+1$ observations constitute the available data set. Associated with these data, we will write M_l^k for the vector $(M_{t_l},\ldots,M_{t_k})^\top$, and m_l^k for the vector of observations $(m_l,\ldots,m_k)^\top$, $0 \leq l \leq k \leq n$.

For the forthcoming analysis, it will be convenient to introduce a number of additional random variables. The indicator random variable for the event that at time $t = 0$ the background process is in state i will be denoted by B_i, that is,

$$B_i = 1_{\{X_0 = i\}}, \quad i = 1,\ldots,d.$$

We also define, for all $k = 1,\ldots,n$ and $i = 1,\ldots,d$, the following random variables corresponding to the k-th interval $\big(t_{k-1}, t_k\big]$:

$Z_i^{[k]}$ = the total amount of time spent in state i by $\{X_t\}$;

$Y_{ij}^{[k]}$ = the total number of state transitions of $\{X_t\}$ from state i to j $(j \neq i)$;

$A_i^{[k]}$ = the total number of arrivals while the background process is in state i.

Finally, in the sequel we will write $X = \{X_t : 0 \leq t \leq t_n\}$, $Y = \sum_{k=1}^{n}\sum_{i=1}^{d}\sum_{j\neq i} Y_{i,j}^{[k]}$ and $A = \{A_i^{[k]} : i = 1,\ldots,d, k = 1,\ldots,n\}$. We see that Y equals the total number of jumps of $\{X_t\}$ in $(0,t_n]$. Let $J_1,\ldots,J_Y$ be the corresponding jump times of $\{X_t\}$, and $J_0 = 0$.

The corresponding states after the jumps will be denoted by $S_0, \ldots, S_Y$, so that $S_l = X_{J_l}$, $l = 0, \ldots, Y$.

2.2.2 Parameter estimation

Our goal is to estimate the unknown parameters of an MMIS process given the number of states d, and observations $m_0, \ldots, m_n$ of $M_{t_0}, \ldots, M_{t_n}$. We consider the setting that the departure rate μ is time-independent and known, and we thus concentrate on estimating the parameter vector $\theta = (\pi_i, q_{ij}, \lambda_i : i, j \in \{1, \ldots, d\},\ j \neq i)^\top$. The estimate will be denoted by $\hat{\theta} = (\hat{\pi}_i, \hat{q}_{ij}, \hat{\lambda}_i : i, j \in \{1, \ldots, d\},\ j \neq i)^\top$.

Let $v = (1, \ldots, 1)^\top$ be a vector of size d. By taking into account the background process $\{X_t\}$ at the observation times and using (2.1) and (2.2), we find that the likelihood function $\mathcal{L}_0$ is given by

$$\begin{aligned}\mathcal{L}_0(\theta \mid M_0^n) &= \mathbb{P}_\theta(M_0^n = m_0^n) \\ &= \sum_{x_0,\ldots,x_n} \mathbb{P}_\theta(M_{t_0} = m_0, X_{t_0} = x_0, \ldots, M_{t_n} = m_n, X_{t_n} = x_n) \\ &= \pi^\top \Big(\prod_{k=1}^{n} P_\Delta(m_{k-1}, m_k) \Big) v. \end{aligned} \tag{2.3}$$

Maximum likelihood estimation based on (2.3) is problematic, since the likelihood is expressed in terms of matrix multiplications. These matrix multiplications appear due to the fact that we observe the process $\{M_t\}$ only at discrete time points and because the process $\{X_t\}$ is unobserved. As indicated above, we will use the EM algorithm to tackle the estimation problem.

2.3 The algorithm

The EM algorithm starts with an initial input value θ^0 and then updates the estimate $\hat{\theta}$ iteratively. Each iteration in the EM algorithm consists of an expectation step and a maximization step, which together produce a parameter update $\tilde{\theta} = (\tilde{\pi}_i, \tilde{q}_{ij}, \tilde{\lambda}_i : i, j \in \{1, \ldots, d\},\ j \neq i)^\top$ from a parameter input $\theta' = (\pi'_i, q'_{ij}, \lambda'_i : i, j \in \{1, \ldots, d\},\ j \neq i)^\top$. The key here is that, instead of maximizing the loglikelihood $\mathcal{L}_0$ based on the observed data, the loglikelihood $\mathcal{L}$ based on a larger data set is maximized. This larger data set, called the complete data set, consists of the observed data and missing data. The missing data can be missing observations, or, as in our case, conveniently chosen unobserved data.

In Section 2.3.1 we describe the expectation and maximization steps and derive expressions for the parameter updates $\tilde{\pi}_i, \tilde{q}_{ij}$ and $\tilde{\lambda}_i$. Sections 2.3.2–2.3.5 elaborate on how to compute these expressions explicitly. Section 2.3.6 summarizes the entire algorithm with which the parameter estimates can be obtained.

2.3.1 Parameter updates

To perform the expectation and maximization steps, we consider (A, X) as the missing data, so that (M_0^n, A, X) is the complete data set. The loglikelihood function of the complete data is then

$$\log \mathcal{L}(\theta | M_0^n, A, X) = \log \mathbb{P}_\theta(M_0^n | A, X) + \log \mathbb{P}_\theta(A, X). \tag{2.4}$$

For the expectation step of the EM algorithm, we have to compute

$$\begin{aligned} &\mathbb{E}_{\theta'}\left[\log \mathcal{L}(\theta \,|\, M_0^n, A, X)\middle| M_0^n = m_0^n\right] \\ &\quad = \mathbb{E}_{\theta'}\left[\log \mathbb{P}_\theta(M_0^n \,|\, A, X)\middle| M_0^n = m_0^n\right] + \mathbb{E}_{\theta'}\left[\log \mathbb{P}_\theta(A, X)\middle| M_0^n = m_0^n\right]. \end{aligned} \tag{2.5}$$

For the maximization step we compute

$$\begin{aligned} \tilde{\theta} &= \arg\max_\theta \mathbb{E}_{\theta'}[\log \mathcal{L}(\theta | M_0^n, A, X) | M_0^n = m_0^n] \\ &= \arg\max_\theta \mathbb{E}_{\theta'}[\log \mathbb{P}_\theta(A, X) | M_0^n = m_0^n]. \end{aligned} \tag{2.6}$$

Note that (2.6) follows from (2.5) because the first term on the right hand side of (2.4) – and hence also of (2.5) – only depends on the known departure rate μ and not on the unknown parameter θ. Hence, given A and X, the arrivaltimes in each state follow a uniform distribution by a well-known property of the homogeneous Poisson process, and therefore $\mathbb{P}_\theta(M_0^n | A, X)$ translates into a probability on the departures only.

To find the parameter update $\tilde{\theta}$ the expectation $\mathbb{E}_{\theta'}[\log \mathbb{P}_\theta(A, X) | M_0^n = m_0^n]$ needs to be computed. We first observe that

$$\mathbb{P}_\theta(A, X) = \mathbb{P}_\theta(A | X)\, \mathbb{P}_\theta(X). \tag{2.7}$$

By using the partition of the interval $(0, t_n]$ into the observation intervals $(t_{k-1}, t_k]$, $k = 1, \ldots, n$, we see that

$$\mathbb{P}_\theta(A \,|\, X) = \prod_{k=1}^{n} \prod_{i=1}^{d} \frac{(\lambda_i Z_i^{[k]})^{A_i^{[k]}}}{(A_i^{[k]})!} e^{-\lambda_i Z_i^{[k]}}. \tag{2.8}$$

For the computation of $\mathbb{P}_\theta(X)$ we do not use this partition, but consider the entire interval $(0, t_n]$. We have

$$\mathbb{P}_\theta(X) = \pi_{S_0} \prod_{y=1}^{Y} \left(q_{S_{y-1} S_y} e^{-q_{S_{y-1}} \left(J_y - J_{y-1}\right)} \right) e^{-q_{S_Y} \left(t_n - J_Y\right)}. \tag{2.9}$$

Combining (2.7), (2.8) and (2.9), and rewriting the obtained expression by aggregating all terms with π_i, all terms with q_{ij} and all terms with λ_i, we find

$$\begin{aligned}\log \mathbb{P}_\theta(A, X) = &\sum_{i=1}^{d} \log(\pi_i) B_i + \sum_i \sum_{j \neq i} \sum_{k=1}^{n} \left(Y_{i,j}^{[k]} \log(q_{ij}) - q_{ij} Z_i^{[k]} \right) \\ &+ \sum_{i=1}^{d} \sum_{k=1}^{n} \left(A_i^{[k]} \log(\lambda_i) - \lambda_i Z_i^{[k]} \right) + \sum_{i=1}^{d} \sum_{k=1}^{n} \left(A_i^{[k]} \log(Z_i^{[k]}) - \log(A_i^{[k]}!) \right).\end{aligned} \tag{2.10}$$

Substituting this result into (2.6) and solving the equation for $\tilde{\theta}$, we obtain the parameter updates

$$\begin{aligned}\tilde{\pi}_i &= \mathbb{E}_{\theta'}[B_i | M_0^n = m_0^n], \\ \tilde{q}_{ij} &= \frac{\sum_{k=1}^{n} \mathbb{E}_{\theta'} \left[Y_{i,j}^{[k]} \middle| M_0^n = m_0^n \right]}{\sum_{k=1}^{n} \mathbb{E}_{\theta'} \left[Z_i^{[k]} \middle| M_0^n = m_0^n \right]}, \\ \tilde{\lambda}_i &= \frac{\sum_{k=1}^{n} \mathbb{E}_{\theta'} \left[A_i^{[k]} \middle| M_0^n = m_0^n \right]}{\sum_{k=1}^{n} \mathbb{E}_{\theta'} \left[Z_i^{[k]} \middle| M_0^n = m_0^n \right]}.\end{aligned} \tag{2.11}$$

In the next three sections, we further elaborate on how to compute these parameter updates explicitly.

2.3.2 Transition probabilities

Before the parameter updates of (2.11) can be computed, some preliminary steps need to be taken. First we show how to obtain approximations of the transition probability matrices defined in (2.2), which will be used in the next steps.

Exact computation of the transition probability matrices is not feasible, since the computation of the transition probabilities would need taking the exponent of the transition rate matrix of $\{M_t, X_t\}$, which is problematic for systems with infinite state space. We therefore approximate our MMIS process by a process in which the population size is bounded from above by a finite number C. In other words, we consider the same system as described above but now with the restriction that there can be at most C individuals in the system. New arrivals are blocked whenever the system is full. Because of this restriction, the system will behave slightly differently, but we can choose C large enough such that $\mathbb{P}(M_t = m)$ is negligible for $m > C$ for all t. The transition probabilities of the original system can then be well approximated by the transition probabilities of the process with population size bounded by C, and we can use

$$P_t(m, m') \approx P_t^C(m, m'), \tag{2.12}$$

where $P_t^C(m, m')$ is the analog of $P_t(m, m')$ for the system with population size bounded

by C.

For the system with population size bounded by C we can find the transition probabilities by taking the exponent of its $d(C+1) \times d(C+1)$ transition rate matrix R^C. This transition rate matrix has a tridiagonal form and is given by

$$R^C = \begin{pmatrix} R_0 & R_1 & & & \\ \mu I_d & R_0 - \mu I_d & \ddots & & \\ & 2\mu I_d & \ddots & R_1 & \\ & & \ddots & R_0 - (C-1)\mu I_d & R_1 \\ & 0 & & C\mu I_d & Q - C\mu I_d \end{pmatrix},$$

where I_d is the $d \times d$ identity matrix, $R_1 = \mathrm{diag}\{\lambda_1, \ldots, \lambda_d\}$ and $R_0 = Q - R_1$. The t-time transition probability matrix is obtained by taking the matrix exponent of $R^C t$. Note that this probability matrix is a composition of the $d \times d$ block matrices $P_t^C(m, m')$ in which, for fixed $0 \leq m, m' \leq C$, the (i, j)-th entry is equal to the transition probability $p_{ij}^C(m, m'; t)$. More specifically, let us define, for $0 \leq i \leq C$, e_i as the $d(C+1) \times d$ matrix which consists of the identity matrix I_d at the $(i+1)$-th block and zeros elsewhere. Then

$$P_t^C(m, m') = e_m^\top \left[e^{R^C t}\right] e_{m'}. \tag{2.13}$$

2.3.3 Forward and backward vectors

We will now introduce the forward and backward vectors that are involved in the EM-algorithm for the MMIS process and show how to compute them. These forward and backward vectors will be used to obtain the conditional expectations in (2.11).

The forward vector $f_{k,\theta}(m, u)$ is defined for $k = 0, \ldots, n$, $m \geq 0$ and $0 \leq u \leq \Delta$, as the vector of length d with i-th entry

$$[f_{k,\theta}(m, u)]_i = \mathbb{P}_\theta(M_0^k = m_0^k, M_{(t_k+u)^-} = m, X_{(t_k+u)^-} = i). \tag{2.14}$$

The backward vector $b_{k,\theta}(m, u)$ is defined for $k = 0, \ldots, n-1$, $m \geq 0$ and $0 \leq u \leq \Delta$, as the vector of length d with i-th entry

$$[b_{k,\theta}(m, u)]_i = \mathbb{P}_\theta(M_k^n = m_k^n | M_{(t_k-u)^+} = m, X_{(t_k-u)^+} = i). \tag{2.15}$$

In the above, the notation M and X with indices $(t_k + u)^-$ and $(t_k - u)^+$ indicates the values of M and X just before time $(t_k + u)$ and just after time $(t_k - u)$, respectively.

To compute $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$ in an efficient way, we first consider the special cases $f_{k,\theta} = f_{k,\theta}(m_k, 0)$ and $b_{k,\theta} = b_{k,\theta}(m_k, 0)$. In view of (2.1) and (2.2) we have for

$i = 1, 2,$

$$[f_{k,\theta}]_i = \sum_{x_0,\ldots,x_{k-1}} \pi_{x_0} \prod_{l=1}^{k} \mathbb{P}_\theta(M_{t_l} = m_l, X_{t_l} = x_l | M_{t_{l-1}} = m_{l-1}, X_{t_{l-1}} = x_{l-1})$$
$$= \Big[\Big(\prod_{l=1}^{k} P_\Delta(m_{l-1}, m_l) \Big)^\top \pi \Big]_i,$$

and

$$[b_{k,\theta}]_i = \sum_{x_{k+1},\ldots,x_n} \prod_{l=k+1}^{n} \mathbb{P}_\theta(M_{t_l} = m_l, X_{t_l} = x_l | M_{t_{l-1}} = m_{l-1}, X_{t_{l-1}} = x_{l-1})$$
$$= \Big[\Big(\prod_{l=k+1}^{n} P_\Delta(m_{l-1}, m_l) \Big) v \Big]_i.$$

Since, for $k = 1, \ldots, n$,

$$f_{k,\theta} = P_\Delta(m_{k-1}, m_k)^\top f_{k-1,\theta}, \quad \text{with initial condition } f_{0,\theta} = \pi, \tag{2.16}$$

and for $k = 0, \ldots, n-1$,

$$b_{k,\theta} = P_\Delta(m_k, m_{k+1}) b_{k+1,\theta}, \quad \text{with initial condition } b_{n,\theta} = v, \tag{2.17}$$

we see that $f_{k,\theta}$ and $b_{k,\theta}$ can be computed recursively. After the computation of $f_{1,\theta}, \ldots, f_{n,\theta}$ and $b_{0,\theta}, \ldots, b_{n-1,\theta}$ by the recurrence relations (2.16) and (2.17), respectively, the general versions $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$ can be computed by

$$f_{k,\theta}(m, u) = P_u(m_k, m)^\top f_{k,\theta},$$

and

$$b_{k,\theta}(m, u) = P_u(m, m_k) b_{k,\theta},$$

which can be evaluated using (2.12) and (2.13).

2.3.4 Conditional expectations

Having seen how to compute the transition probability matrices of (2.2) and the forward and backward vectors, which are necessary tools for the computation of the conditional expectations in (2.11), we are ready to derive expressions for these conditional expectations in terms of the $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$. Below we will make frequent use of the

definitions (2.14) and (2.15) of these forward and backward vectors.

As a start, we note that

$$\mathbb{E}_{\theta'}\left[B_i \middle| M_0^n = m_0^n\right] = \frac{1}{\mathbb{P}_{\theta'}(M_0^n = m_0^n)}\,\mathbb{E}_{\theta'}\left[B_i\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right]. \tag{2.18}$$

In (2.18) B_i can be replaced by $Y_{i,j}^{[k]}$, $Z_i^{[k]}$ or $A_i^{[k]}$, to obtain an analogous relationship for the other conditional expectations in (2.11). Since $\mathbb{P}_{\theta'}(M_0^n = m_0^n)$ on the right-hand side of these relations is given by (2.3), we only need expressions in terms of the forward and backward vectors for the expectations $\mathbb{E}_{\theta'}\left[B_i\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right]$, $\mathbb{E}_{\theta'}\left[Y_{i,j}^{[k]}\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right]$, $\mathbb{E}_{\theta'}\left[Z_i^{[k]}\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right]$ and $\mathbb{E}_{\theta'}\left[A_i^{[k]}\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right]$.

First, we observe that $\mathbb{E}_{\theta'}\left[B_i\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right]$ is simply computed by

$$\mathbb{E}_{\theta'}\left[B_i\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right] = \mathbb{P}_{\theta'}(X_0 = i, M_0^n = m_0^n) = \pi_i'[b_{0,\theta'}]_i. \tag{2.19}$$

Second, for $\mathbb{E}_{\theta'}\left[Y_{i,j}^{[k]}\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right]$, we have that

$$\begin{aligned}
&\mathbb{E}_{\theta'}\left[Y_{i,j}^{[k]}\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right] \\
&\quad= \int_0^\Delta \mathbb{P}_{\theta'}(X_{(t_{k-1}+\tau)^-} = i, X_{(t_{k-1}+\tau)^+} = j, M_0^n = m_0^n)d\tau \\
&\quad= \int_0^\Delta \sum_{m=0}^{\infty} \mathbb{P}_{\theta'}(M_k^n = m_k^n | X_{(t_{k-1}+\tau)^+} = j, M_{(t_{k-1}+\tau)^+} = m) q_{ij}' \\
&\qquad \mathbb{P}_{\theta'}(M_0^{k-1} = m_0^{k-1}, X_{(t_{k-1}+\tau)^-} = i, M_{(t_{k-1}+\tau)^-} = m)d\tau,
\end{aligned} \tag{2.20}$$

where in the second step we conditioned on the population size at time τ, and used the Markov property. The last integral in (2.20) can be rewritten in terms of the forward and backward vectors $f_{k,\theta}(m,u)$ and $b_{k,\theta}(m,u)$, to obtain

$$\mathbb{E}_{\theta'}\left[Y_{i,j}^{[k]}\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right] = \int_0^\Delta \sum_{m=0}^{\infty} [f_{k-1,\theta'}(m,\tau)]_i q_{ij}' [b_{k,\theta'}(m,\Delta-\tau)]_j d\tau. \tag{2.21}$$

In a similar way, we have for $\mathbb{E}_{\theta'}\left[Z_i^{[k]}\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right]$,

$$\begin{aligned}
\mathbb{E}_{\theta'}\left[Z_i^{[k]}\,\mathbf{1}_{\{M_0^n = m_0^n\}}\right] &= \int_0^\Delta \mathbb{P}_{\theta'}(X_{t_{k-1}+\tau} = i, M_0^n = m_0^n)d\tau \\
&= \int_0^\Delta \sum_{m=0}^{\infty} \mathbb{P}_{\theta'}(M_0^{k-1} = m_0^{k-1}, M_{t_{k-1}+\tau} = m, X_{t_{k-1}+\tau} = i) \\
&\qquad \mathbb{P}_{\theta'}(M_k^n = m_k^n | M_{t_{k-1}+\tau} = m, X_{t_{k-1}+\tau} = i)d\tau.
\end{aligned} \tag{2.22}$$

Rewriting (2.22) in terms of $f_{k,\theta}(m,u)$ and $b_{k,\theta}(m,u)$, we get

$$\mathbb{E}_{\theta'}\left[Z_i^{[k]}\,\mathbf{1}_{\{M_0^n=m_0^n\}}\right] = \int_0^\Delta \sum_{m=0}^{\infty} [f_{k-1,\theta'}(m,\tau)]_i [b_{k,\theta'}(m,\Delta-\tau)]_i d\tau. \tag{2.23}$$

Lastly, we find the expression for $\mathbb{E}_{\theta'}\left[A_i^{[k]}\,\mathbf{1}_{\{M_0^n=m_0^n\}}\right]$. Let $\mathcal{A}_t$ be the event that an arrival occurs at time $t>0$. Then

$$\begin{aligned}
&\mathbb{E}_{\theta'}\left[A_i^{[k]}\,\mathbf{1}_{\{M_0^n=m_0^n\}}\right] \\
&\quad= \int_0^\Delta \mathbb{P}_{\theta'}(\mathcal{A}_\tau, X_\tau = i, M_0^n = m_0^n)d\tau \\
&\quad= \int_0^\Delta \sum_{m=0}^{\infty} \mathbb{P}_{\theta'}(M_0^{k-1} = m_0^{k-1}, M_{(t_{k-1}+\tau)^-} = m, X_{(t_{k-1}+\tau)^-} = i) \\
&\qquad \mathbb{P}_{\theta'}(M_{(t_{k-1}+\tau)^+} = m+1, X_{(t_{k-1}+\tau)^+} = i \,|\, M_{(t_{k-1}+\tau)^-} = m, X_{(t_{k-1}+\tau)^-} = i) \\
&\qquad \mathbb{P}_{\theta'}(M_k^n = m_k^n \,|\, M_{(t_{k-1}+\tau)^+} = m+1, X_{(t_{k-1}+\tau)^+} = i)\, d\tau,
\end{aligned}$$

and rewriting in terms of $f_{k,\theta}(m,u)$ and $b_{k,\theta}(m,u)$ yields

$$\mathbb{E}_{\theta'}\left[A_i^{[k]}\,\mathbf{1}_{\{M_0^n=m_0^n\}}\right] = \int_0^\Delta \sum_{m=0}^{\infty} [f_{k-1,\theta'}(m,\tau)]_i \lambda_i' [b_{k,\theta'}(m+1,\Delta-\tau)]_i \, d\tau. \tag{2.24}$$

We note that the obtained results (2.21), (2.23) and (2.24) are very similar, but that there are some minor but crucial differences. The entries of the forward and backward vectors differ per expression and the variable in the backward vector is equal to $m+1$ in (2.24) in contrast to m in (2.21) and (2.23).

2.3.5 Differential equations method

In order to use equations (2.21), (2.23) and (2.24) for computing the parameter updates of (2.11), we need a method to compute the integrals on their right-hand sides in an efficient way. For this we propose a differential equations method.

Because $[f_{k-1,\theta'}(m,\tau)]_i$ is negligible for $m > C$, we may truncate the sums in (2.21), (2.23) and (2.24) at the finite bound C. Next, we introduce, for $0 \le m, m' \le C$ and $t>0$, the $d \times d$ matrix

$$G_{k,\theta}(m,m',t) = \int_0^t b_{k,\theta}(m', t-\tau) f_{k-1,\theta}(m,\tau)^\top d\tau, \tag{2.25}$$

so that the conditional expectations in (2.11) become equal to

$$\mathbb{E}_{\theta'}\left[Y_{i,j}^{[k]}\middle|M_0^n = m_0^n\right] = \frac{q'_{ij}}{\mathbb{P}_{\theta'}(M_0^n = m_0^n)} \sum_{m=0}^{C} [G_{k,\theta'}(m, m, \Delta)]_{i,j}, \tag{2.26}$$

$$\mathbb{E}_{\theta'}\left[Z_i^{[k]}\middle|M_0^n = m_0^n\right] = \frac{1}{\mathbb{P}_{\theta'}(M_0^n = m_0^n)} \sum_{m=0}^{C} [G_{k,\theta'}(m, m, \Delta)]_{i,i}, \tag{2.27}$$

$$\mathbb{E}_{\theta'}\left[A_i^{[k]}\middle|M_0^n = m_0^n\right] = \frac{\lambda'_i}{\mathbb{P}_{\theta'}(M_0^n = m_0^n)} \sum_{m=0}^{C-1} [G_{k,\theta'}(m, m+1, \Delta)]_{i,i}. \tag{2.28}$$

To facilitate the computation of (2.26)–(2.28), we derive a system of differential equations for $G_{k,\theta}(m, m', t)$.

For $f_{k,\theta}(m, u)$ and $b_{k,\theta}(m, u)$ a system of differential equations can be easily obtained from the derivatives of the corresponding transition probabilities while making use of (2.13), (2.14) and (2.15). For the forward vector, this yields for $1 \leq m \leq C-1$,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}u} f_{k,\theta}(0, u) &= R_0^\top f_{k,\theta}(0, u) + \mu f_{k,\theta}(1, u), \\
\frac{\mathrm{d}}{\mathrm{d}u} f_{k,\theta}(m, u) &= R_1^\top f_{k,\theta}(m-1, u) + R_0^\top f_{k,\theta}(m, u) \\
&\quad + (m+1)\mu f_{k,\theta}(m+1, u) - m\mu f_{k,\theta}(m, u), \\
\frac{\mathrm{d}}{\mathrm{d}u} f_{k,\theta}(C, u) &= Q^\top f_{k,\theta}(C, u) + R_1^\top f_{k,\theta}(C-1, u) - C\mu f_{k,\theta}(C, u),
\end{aligned} \tag{2.29}$$

with initial condition $f_{0,\theta}(m_0, 0) = \pi$, and for the backward vector we find for $0 \leq m \leq C-1$,

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}u} b_{k,\theta}(m, u) &= R_1 b_{k,\theta}(m+1, u) + R_0 b_{k,\theta}(m, u) \\
&\quad + m\mu b_{k,\theta}(m-1, u) - m\mu b_{k,\theta}(m, u), \\
\frac{\mathrm{d}}{\mathrm{d}u} b_{k,\theta}(C, u) &= Q b_{k,\theta}(C, u) + C\mu b_{k,\theta}(C-1, u) - C\mu b_{k,\theta}(C, u),
\end{aligned} \tag{2.30}$$

with initial condition $b_{n,\theta}(m_n, 0) = v$. Furthermore, from (2.25) we get

$$\frac{\mathrm{d}}{\mathrm{d}t} G_{k,\theta}(m, m', t) = \int_0^t \frac{\mathrm{d}}{\mathrm{d}t} b_{k,\theta}(m', t-\tau) f_{k-1,\theta}(m, \tau)^\top d\tau + b_{k,\theta}(m', 0) f_{k-1,\theta}(m, t)^\top. \tag{2.31}$$

Combining the differential equations for the backward vectors from (2.30) and (2.31), we

obtain, for $m = 0, \ldots, C$, the following system of differential equation for $G_{k,\theta}(m, m', t)$:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} G_{k,\theta}(m, m', t) &= R_1 G_{k,\theta}(m, m'+1, t) + R_0 G_{k,\theta}(m, m', t) + m'\mu G_{k,\theta}(m, m'-1, t) \\
&\quad - m'\mu G_{k,\theta}(m, m', t) + b_{k,\theta}(m', 0) f_{k-1,\theta}(m, t)^\top, \quad 0 \leq m' \leq C-1 \\
\frac{\mathrm{d}}{\mathrm{d}t} G_{k,\theta}(m, C, t) &= Q G_{k,\theta}(m, C, t) + C\mu G_{k,\theta}(m, C-1, t) \\
&\quad - C\mu G_{k,\theta}(m, C, t) + b_{k,\theta}(C, 0) f_{k-1,\theta}(m, t)^\top.
\end{aligned} \tag{2.32}$$

Note that these differential equations contain the term $b_{k,\theta}(m', 0) f_{k-1,\theta}(m, t)^\top$, in which $f_{k-1,\theta}(m, t)$ depends on the variable of differentiation t. Therefore, the differential equations for the forward vectors in (2.29) are needed to solve the differential equations for $G_{k,\theta}(m, m', t)$.

Analyzing the system of differential equations in (2.32) a bit further, we observe that the system can be split into d independent systems of differential equations. For this, we consider each column of the matrix $G_{k,\theta}(m, m', t)$ separately. Let the j-th column of $G_{k,\theta}(m, m', t)$ be denoted by $[G_{k,\theta}(m, m', t)]_j$. Then for each $j = 1, \ldots, d$, we stack the j-th columns of the matrices $G_{k,\theta}(0, m', t), \ldots, G_{k,\theta}(C, m', t)$, into $d(C+1)$-dimensional vectors of the form

$$G_{k,\theta}(m', t)_j = \begin{pmatrix} [G_{k,\theta}(0, m', t)]_j \\ [G_{k,\theta}(1, m', t)]_j \\ \vdots \\ [G_{k,\theta}(C, m', t)]_j \end{pmatrix}.$$

From (2.32) it follows that

$$\frac{\mathrm{d}}{\mathrm{d}t} G_{k,\theta}(m', t)_j = R^C G_{k,\theta}(m', t)_j + c_{k,\theta}(t)_j, \tag{2.33}$$

where $c_{k,\theta}(t)_j$ is a vector containing $[f_{k-1,\theta}(m', t)]_j [b_{k,\theta}]_1$ and $[f_{k-1,\theta}(m', t)]_j [b_{k,\theta}]_2$ at its entries $2m'+1$ and $2m'+2$ respectively, and zeros elsewhere. We note that for each $j = 1, \ldots, d$, (2.33) is a linear system of differential equations for which the solution is equal to

$$G_{k,\theta}(m', t)_j = \int_0^t e^{R^C(t-s)} c_{k,\theta}(t)_j ds.$$

2.3.6 Summarized algorithm

In Sections 2.3.1–2.3.5, we elaborated on the expectation and maximization steps to find the parameter updates. Iteratively repeating the above obtained building blocks for computing the parameter updates results in the complete algorithm for obtaining estimates of the arrival parameters. The algorithm is presented below.

Algorithm

1 Determine initial values $\theta^0 = (\pi_i^0, q_{ij}^0, \lambda_i^0 : i, j \in \{1, \dots, d\}, j \neq i)^\top$ and set $\theta' = \theta^0$.

2 Compute $f_{k,\theta'}$ for $k = 1, \dots, n-1$ and $b_{k,\theta'}$ for $k = 0, \dots, n-1$ by recurrence relations (2.16) and (2.17).

3 Compute $G_{k,\theta'}(m, m, \Delta)$ and $G_{k,\theta'}(m, m+1, \Delta)$ for all $m = 0, \dots, C$ and $k = 1, \dots, n$ by solving differential equations (2.32).

4 Compute conditional expectations (2.19), (2.26), (2.27) and (2.28) for all $k = 1, \dots, n$.

5 Compute parameter updates according to (2.11).

6 If stopping criterion is not satisfied, set $\theta' = \tilde{\theta}$ and go to step 2; else stop algorithm and use final parameter update $\tilde{\theta}$ as parameter estimate $\hat{\theta} = (\hat{\pi}_i, \hat{q}_{ij}, \hat{\lambda}_i : i, j \in \{1, \dots, d\},\ j \neq i)^\top$.

Remark 1. The stopping criterion for the algorithm can be chosen in different ways. As proposed in [50], a reasonable choice is to let the stopping criterion depend on the difference in the loglikelihood functions based on the observed data. In this case the stopping criterion would be given by

$$\left| \log \mathcal{L}_0(\tilde{\theta} \,|\, M_0^n = m_0^n) - \log \mathcal{L}_0(\theta' \,|\, M_0^n = m_0^n) \right| < \varepsilon, \tag{2.34}$$

where ε can be chosen arbitrarily small. Another possibility is to let the stopping criterion depend on the difference between the obtained parameter updates. The stopping criterion would then be

$$\frac{\|\tilde{\theta} - \theta'\|}{\|\theta'\|} < \varepsilon.$$

Remark 2. The observation times $t_0, \dots, t_n$ are defined as equidistant time points with $t_k = k\Delta$. However, the current approach can be applied with general, non-equidistant time points as well. For this, define the sequence $\Delta_1, \dots, \Delta_n$ of lengths of the observation intervals, hence $\Delta_k = t_k - t_{k-1}$ and $t_k = \sum_{i=1}^k \Delta_i$. The parameter estimates in (2.11) will stay the same, but the integrals in (2.21), (2.23) and (2.24) will have upper bound Δ_k instead of Δ, and hence, the $G_{k,\theta'}$ matrices in (2.26)–(2.28) will depend on Δ_k instead of Δ. The algorithm in Section 2.3.6 will remain completely the same.

Remark 3. The main differences between the steps in the algorithm above and the one presented in [50] are the following. In our algorithm the forward and backward vectors featuring in step 2 are redefined in terms of population size (instead of arrivals), and we need a different method to obtain the transition probabilities required to compute these vectors. Additionally, the differential equations appearing in step 3 are more involved, since the $G_{k,\theta}(m, m', t)$ matrices here are defined in terms of population size as well, and therefore require a different solution method. Moreover, our model includes departures and our algorithm enables the estimation of the departure rate (see Section 2.5 below), which is not considered in [50].
Remark 4. With the algorithm not only point estimates, but also bootstrap confidence intervals can be obtained, see Section 2.4.1 below.

2.4 Simulations

In this section, we investigate the accuracy of the proposed algorithm by means of a simulation study. The algorithm was applied to several simulated data sets with varying values of the model parameters, including the time between two observations, Δ, and the sample size n. We simulated the MMIS process with two states, that is $d = 2$, and considered examples in the relevant regime where the background process is relatively slow with respect to the arrival process. If the background process is too fast, the modulated arrival process will be averaged to a Poisson process with a homogeneous rate equal to $\lambda_\infty = \pi_1\lambda_1 + \pi_2\lambda_2$ (see [4]), and as a consequence the modulation will not be detectable from data on the population size.

The algorithm was implemented in MATLAB with the likelihood-based stopping criterion (2.34). Choosing the initial values is a pragmatic procedure and depends on the data setting. Here, the effect of the choice on the initial value for π is negligible, since the parameter updates for π quickly converge to a 0-1 vector. For this reason, the initial value π^0 was set to $(0.5, 0.5)$, and the results on $\hat{\pi}$ were omitted in this section. We used a rough guess on the trace of the background process in combination with moment estimators to find the initial values for the other parameters. There is no crucial difference between the implementation of the algorithm for $d > 2$ and for $d = 2$. The analysis presented in Section 2.3.5 reveals that as d increases by one, only the length of the vectors $G_{k,\theta}(m', t)_i$ and the size of the matrix R^C change, both by $C + 1$. Importantly, in the vast majority of real life processes for which an MMPP is an appropriate model, the number of states is low, and typically 2.

In Section 2.4.1, we discuss the influence of Δ and n on the parameter estimates by varying the values of Δ and n. In Section 2.4.2, we explore the influence of the timescale of the background process $\{X_t\}$ on the parameter estimates by varying the values of the parameters q_1 and q_2.

2.4.1 Influence of Δ and n

We considered the MMIS process with parameter values $\pi = (1, 0)^\top$, $q_1 = 0.3$, $q_2 = 0.9$, $\lambda_1 = 4$, $\lambda_2 = 18$, and $\mu = 0.6$. We simulated 100 times the complete path up to time $T = t_n = n\Delta$ of the background process $\{X_t\}$ and the corresponding population process $\{M_t\}$ with these parameter values. From this we computed for each of the 100 simulations for various values of Δ and n the realization of the data vector $(M_{t_0}, \ldots, M_{t_n})$, which corresponds to the available data if one would observe the number of individuals at times $t_0, \ldots, t_n$ only. To investigate the influence of the interval length Δ, we fixed the total observation time $T = 100$, and considered $\Delta = 0.1$, $\Delta = 0.05$ and $\Delta = 0.025$. To investigate the influence of the number of observations n, we fixed $\Delta = 0.05$ and chose $n = 500$, $n = 1000$, $n = 2000$, $n = 3000$ and $n = 4000$. Because for $n = 2000$ we could use the data vectors that were already computed from the 100 simulations for the combination $T = 100$ and $\Delta = 0.05$, from each simulation with $\Delta = 0.05$ four additional data vectors had to be generated for the other values of n.

Results

The results of the first part of the study, where we considered the three different values of Δ, are presented in Table 2.1. This table shows for each Δ (rows) and each parameter (columns), the mean of the 100 estimates together with the corresponding standard deviation between brackets. All rows are quite similar, from which we can conclude that $\Delta = 0.1$ is already small enough to obtain estimates which lie close to the true parameter values. However, as Δ decreases there is a small decrease in the standard deviations, hence the estimates become more accurate as Δ decreases.

n	Δ	q_1	q_2	λ_1	λ_2
1000	0.1	0.321 (0.105)	0.955 (0.370)	3.883 (0.372)	17.952 (1.528)
2000	0.05	0.321 (0.101)	0.954 (0.346)	3.902 (0.344)	17.911 (1.416)
4000	0.025	0.317 (0.089)	0.958 (0.320)	3.945 (0.323)	17.960 (1.346)

Table 2.1: Mean of estimates of 100 data sets, with corresponding standard deviation between brackets. True parameter values: $q_1 = 0.3, q_2 = 0.9, \lambda_1 = 4, \lambda_2 = 18$.

For the second part of the study, where we examined increasing values of n, the results are shown in Table 2.2 and Figures 2.1–2.4. Table 2.2 contains for each sample size (rows) and for each parameter (columns), the mean values of the 100 estimates together with the corresponding standard deviation between brackets. Figures 2.1–2.4 show histograms of the 100 estimates for the parameters q_1, q_2, λ_1 and λ_2, respectively, where each figure contains five histograms corresponding to the five different values of n. We see from Table 2.2 that the means of the estimates lie closer to the true parameter values for larger values of n. Furthermore, the standard deviations decrease as n increases, which means that the estimates become more accurate when n gets larger. The decrease in standard deviation is also visible in the histograms. Each figure shows that the estimates are concentrated

around the true parameter value, but their standard deviation clearly decreases as n becomes larger. In addition, the histograms look more and more bell-shaped, which is indicative of the distributions of the estimators becoming approximately normal when n increases.

n	q_1	q_2	λ_1	λ_2
500	0.445 (0.441)	1.205 (0.897)	3.899 (0.847)	17.270 (3.780)
1000	0.345 (0.152)	1.026 (0.556)	3.918 (0.483)	17.751 (2.004)
2000	0.321 (0.101)	0.954 (0.346)	3.902 (0.344)	17.911 (1.416)
3000	0.315 (0.083)	0.935 (0.281)	3.942 (0.266)	18.028 (1.224)
4000	0.316 (0.076)	0.940 (0.228)	3.969 (0.236)	18.087 (1.044)

Table 2.2: Mean of estimates of 100 data sets, with corresponding standard deviation between brackets for $\Delta = 0.05$. True parameter values: $q_1 = 0.3, q_2 = 0.9, \lambda_1 = 4, \lambda_2 = 18$.

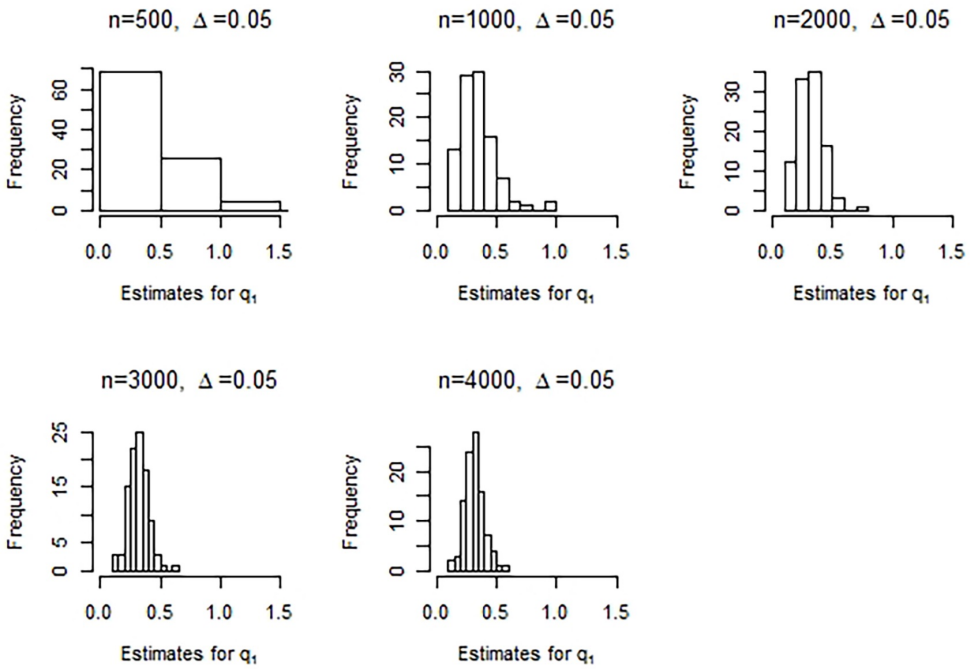

Figure 2.1: Histograms of the obtained estimates for q_1, with n increasing from left to right.

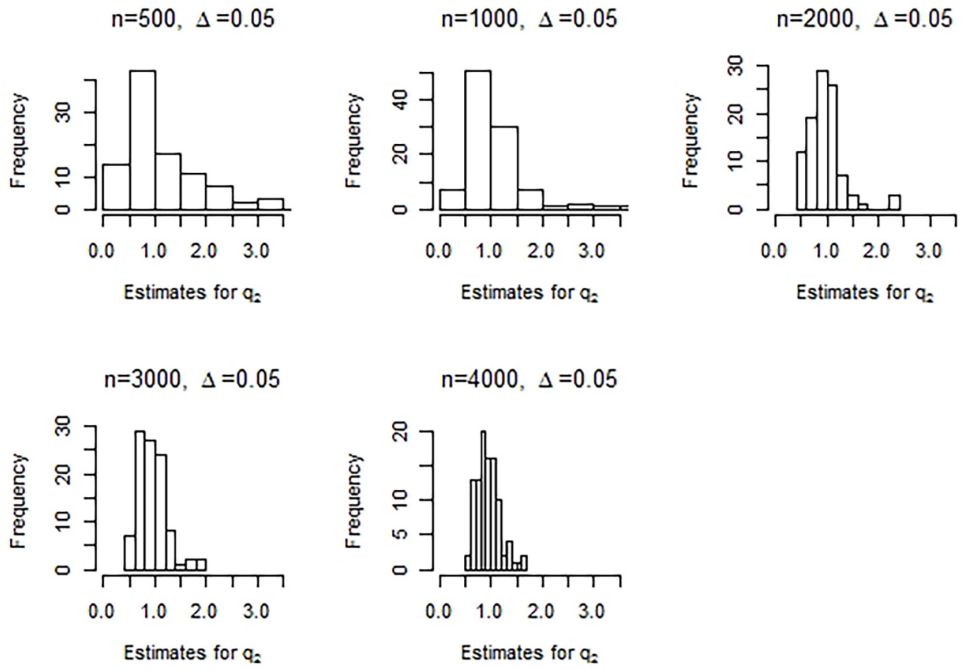

Figure 2.2: Histograms of the obtained estimates for q_2, with n increasing from left to right.

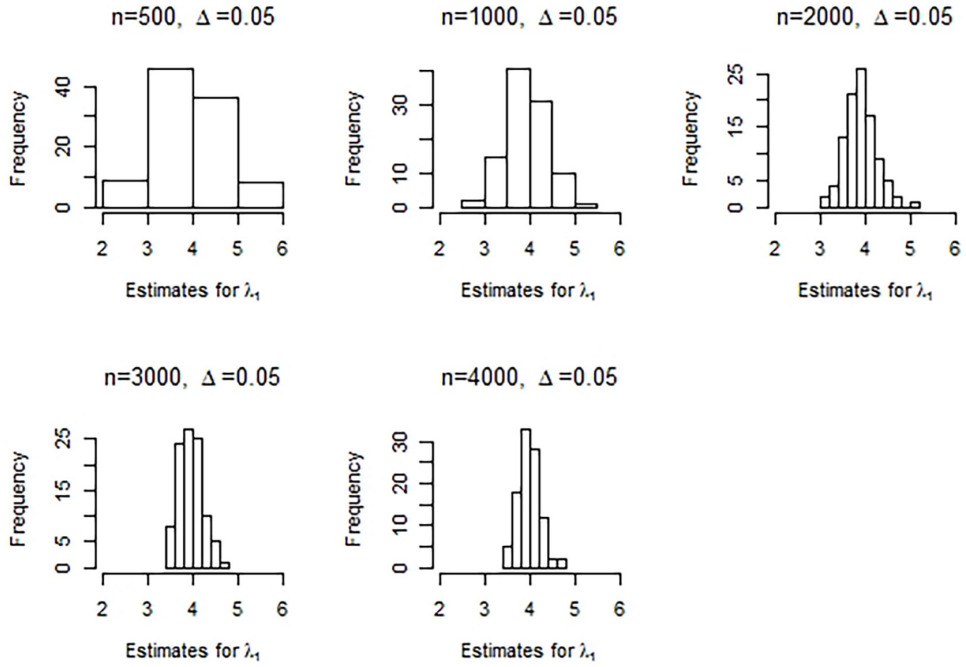

Figure 2.3: Histograms of the obtained estimates for λ_1, with n increasing from left to right.

Bootstrap confidence intervals

We note that this kind of simulation can also be used to construct bootstrap confidence intervals from a real data set. Suppose that a real data set is available with sample size n and interval length Δ, and that with the estimation algorithm the parameter estimates $\hat{q}_1$, $\hat{q}_2$, $\hat{\lambda}_1$ and $\hat{\lambda}_2$ have been obtained. Bootstrap confidence intervals for these parameters can then be computed by similar simulations as above in the following way. Choose $B > 0$ large, for example $B = 1000$, and simulate B new data sets with sample size n and interval length Δ using the parameter values $\hat{q}_1$, $\hat{q}_2$, $\hat{\lambda}_1$ and $\hat{\lambda}_2$. Compute for each simulated data set the corresponding parameter estimates using the estimation algorithm. This yields for each parameter, B bootstrap estimates. These, together with the original estimate, can then be used to construct the confidence interval. As our results above illustrate, the larger the sample size n, the more accurate the parameter estimates will be. Hence, the larger the sample size n, the smaller the confidence intervals will be.

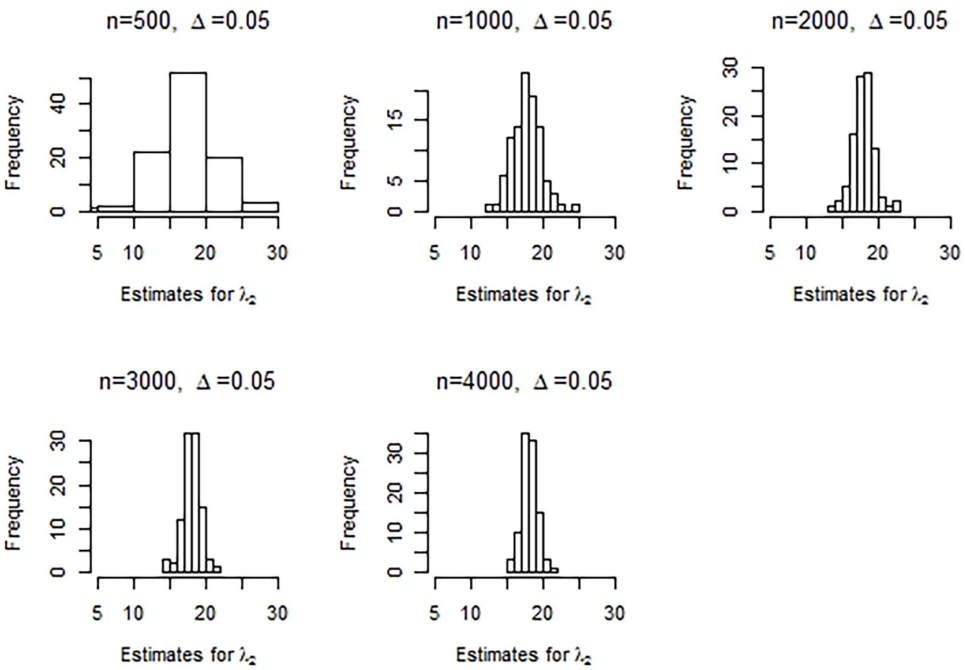

Figure 2.4: Histograms of the obtained estimates for λ_2, with n increasing from left to right.

2.4.2 Influence of the speed of the background process

We now fix $\Delta = 0.05$ and $n = 4000$ and consider the same parameter values as in the previous section, but speed up the background process by varying the values of q_1 and q_2.

We investigated the following four scenarios, where we kept the ratio $q_2/q_1 = 3$ fixed:

1. $q_1 = 0.1, q_2 = 0.3$;
2. $q_1 = 0.3, q_2 = 0.9$;
3. $q_1 = 0.5, q_2 = 1.5$;
4. $q_1 = 1.5, q_2 = 4.5$.

Results

Table 2.3 shows, for each setting (rows) and for each parameter (columns), the mean values of the 100 estimates together with the corresponding standard deviation between brackets. The mean estimates lie close to the true parameter values for each of the four settings. However, the standard deviations clearly increase as the speed of the background process becomes higher (q_1 and q_2 increase). The slower the background process, the easier it is for the algorithm to distinguish the two states and the more accurate the estimates become.

	q_1	q_2	λ_1	λ_2
1.	0.108 (0.037)	0.321 (0.105)	3.967 (0.194)	17.953 (0.718)
2.	0.316 (0.076)	0.940 (0.228)	3.968 (0.236)	18.087 (1.044)
3.	0.528 (0.131)	1.555 (0.343)	3.934 (0.299)	18.036 (1.189)
4.	1.535 (0.396)	4.485 (1.150)	3.897 (0.472)	17.964 (1.738)

Table 2.3: Mean of estimates of 100 data sets, with corresponding standard deviation between brackets, for four settings of the parameter values with $\Delta = 0.05$ and $n = 4000$.

2.5 Estimation of the departure rate μ

Let the departure rate μ now be an unknown parameter which we also want to estimate. In this case the unknown parameter vector is $\theta = (\pi_i, q_{ij}, \lambda_i, \mu : i, j \in \{1, \ldots, d\}, j \neq i)^\top$. In many situations an appropriate assumption would be that an arriving individual does not leave the system in the same observation interval as in which it arrived. We will show that under this assumption the parameter μ can be estimated along with the other parameters of the system in a similar way to the one in Section 2.3.1. Again, let M_0^n be the observed vector and (A, X) be the missing data. For the purpose of this section we start with rewriting the loglikelihood function of the complete data by conditioning on X only, instead of on A and X as we did in (2.4). We have

$$\log \mathcal{L}(\theta | M_0^n, A, X) = \log \mathbb{P}_\theta(M_0^n, A | X) + \log \mathbb{P}_\theta(X). \tag{2.35}$$

In Section 2.3.1, $\mathbb{P}_\theta(X)$ is computed, so the second term on the right hand side of (2.35) can be rewritten using (2.9). However, to find an expression for the first term, a few minor computations need to be made. We use the partition of the interval $(0, t_n]$ into the observation intervals $(t_{k-1}, t_k], k = 1, \ldots, n$, to obtain

$$\begin{aligned}
&\log \mathbb{P}_\theta(M_0^n, A|X) \\
&\quad = \log\left(\prod_{k=1}^{n} \mathbb{P}_\theta(M_k, A_1^{[k]}, \ldots, A_d^{[k]}|M_{k-1}, X)\right) \\
&\quad = \log\left(\prod_{k=1}^{n} \mathbb{P}_\theta(M_k|A_1^{[k]}, \ldots, A_d^{[k]}, M_{k-1}, X) \cdot \mathbb{P}_\theta(A_1^{[k]}|X) \cdot \ldots \cdot \mathbb{P}_\theta(A_d^{[k]}|X)\right) \\
&\quad = \sum_{k=1}^{n} \log \mathbb{P}_\theta(M_k|A_1^{[k]}, \ldots, A_d^{[k]}, M_{k-1}, X) + \sum_{k=1}^{n}\sum_{i=1}^{d} \log \mathbb{P}_\theta(A_i^{[k]}|X).
\end{aligned} \tag{2.36}$$

First, we note that the probability in the first term on the right hand side of (2.36) converts into a probability on the number of departures in the k-th interval $(t_{k-1}, t_k]$. By the additional assumption, newly arrived individuals in $(t_{k-1}, t_k]$ cannot leave the system in this interval. Therefore, only individuals that are already present in the system at time t_{k-1} can leave the system in $(t_{k-1}, t_k]$. We thus have

$$\mathbb{P}_\theta(M_k|A_1^{[k]}, \ldots, A_d^{[k]}, M_{k-1}, X) = \binom{M_{k-1}}{D_k}(1 - e^{-\mu\Delta})^{D_k}(e^{-\mu\Delta})^{M_{k-1}-D_k}, \tag{2.37}$$

for all $0 \leq D_k \leq M_{k-1}$ and zero otherwise. Here $D_k = M_{k-1} - M_k + \sum_{i=1}^{d} A_i^{[k]}$, the number of departures in the k-th interval $(t_{k-1}, t_k]$. Next, we observe that for the second term on the right hand side of (2.36) it holds that

$$\mathbb{P}_\theta(A_i^{[k]}|X) = \frac{(\lambda_i Z_i^{[k]})^{A_i^{[k]}}}{(A_i^{[k]})!} e^{-\lambda_i Z_i^{[k]}}, \quad i = 1, \ldots, d. \tag{2.38}$$

By combining (2.37) and (2.38), (2.36) becomes

$$\begin{aligned}
&\log \mathbb{P}_\theta(M_0^n, A|X) \\
&\quad = \sum_{k=1}^{n}\Bigg[\log\binom{M_{k-1}}{D_k} + D_k \log(1 - e^{-\mu\Delta}) - \mu\Delta(M_{k-1} - D_k) \\
&\qquad + \sum_{i=1}^{d} A_i^{[k]} \log(\lambda_i) + A_i^{[k]} \log(Z_i^{[k]}) - \log(A_i^{[k]}!) - \lambda_i Z_i^{[k]}\Bigg], \quad 0 \leq D_k \leq M_{k-1}.
\end{aligned} \tag{2.39}$$

Using (2.9), (2.35) and (2.39), we can rewrite the loglikelihood function of the complete data by aggregating all terms with π_i, all terms with q_{ij}, all terms with λ_i and all terms

with μ. This yields

$$\begin{aligned}
&\log \mathcal{L}(\theta|M_0^n, A, X) \\
&\quad = \sum_{i=1}^{d} \log(\pi_i) B_i + \sum_i \sum_{j \neq i} \sum_{k=1}^{n} \left(Y_{i,j}^{[k]} \log(q_{ij}) - q_{ij} Z_i^{[k]} \right) \\
&\quad + \sum_{i=1}^{d} \sum_{k=1}^{n} \left(A_i^{[k]} \log(\lambda_i) - \lambda_i Z_i^{[k]} \right) + \sum_{i=1}^{d} \sum_{k=1}^{n} \left(A_i^{[k]} \log(Z_i^{[k]}) - \log(A_i^{[k]}!) \right) \\
&\quad + \sum_{k=1}^{n} \left(\log \binom{M_{k-1}}{D_k} + D_k \log(1 - e^{-\mu\Delta}) - \mu\Delta(M_{k-1} - D_k) \right).
\end{aligned} \tag{2.40}$$

It can be seen that (2.40) and (2.10) are precisely the same except for the additional last term on the right hand side of (2.40), which depends on μ and does not depend on the other parameters. Hence, we obtain the same parameter updates for π_i, q_{ij} and λ_i as in Section 2.3.1, while the additional parameter update for μ will be based on the last term on the right hand side of (2.40).

Let the parameter update for μ be denoted by $\tilde{\mu}$. Since not only the first four terms on the right hand side of (2.40), but also $\log \binom{M_{k-1}}{D_k}$ does not depend on μ, we see from (2.40) that

$$\begin{aligned}
\tilde{\mu} &= \arg\max_{\mu} \mathbb{E}_{\theta'}[\log \mathcal{L}(\theta|M_0^n, A, X)|M_0^n = m_0^n] \\
&= \arg\max_{\mu} \mathbb{E}_{\theta'} \left[\sum_{k=1}^{n} \left(\log \binom{M_{k-1}}{D_k} + D_k \log(1 - e^{-\mu\Delta}) - \mu\Delta(M_{k-1} - D_k) \right) \middle| M_0^n = m_0^n \right] \\
&= \arg\max_{\mu} \left(\log(1 - e^{-\mu\Delta}) \sum_{k=1}^{n} \mathbb{E}_{\theta'}[D_k|M_0^n = m_0^n] - \mu\Delta \sum_{k=1}^{n} \mathbb{E}_{\theta'}[(M_{k-1} - D_k)|M_0^n = m_0^n] \right).
\end{aligned} \tag{2.41}$$

Solving equation (2.41) for μ and using $D_k = M_{k-1} - M_k + \sum_{i=1}^{d} A_i^{[k]}$ yields the parameter update

$$\tilde{\mu} = \frac{1}{\Delta} \log \left(\frac{\sum_{k=1}^{n} m_{k-1}}{\sum_{k=1}^{n} m_k - \sum_{i=1}^{d} \sum_{k=1}^{n} \mathbb{E}_{\theta'} \left[A_i^{[k]} \middle| M_0^n = m_0^n \right]} \right). \tag{2.42}$$

Note that this parameter update depends on $\sum_{k=1}^{n} \mathbb{E}_{\theta'} \left[A_i^{[k]} \middle| M_0^n = m_0^n \right], i = 1, \ldots, d$, which are already computed in updating the parameters λ_i, see (2.11). Hence, the parameter update of μ can be obtained in addition to the other parameter updates without much extra computational effort. The algorithm for finding maximum likelihood estimates for the new parameter vector $\theta = (\pi_i, q_{ij}, \lambda_i, \mu : i, j \in \{1, \ldots, d\}, j \neq i)^\top$ is the same as that in Section 2.3.6 with the single addition of computing $\tilde{\mu}$ according to (2.42) in step 5 of

the algorithm.

$\mu^0 = 0.1$

n	q_1	q_2	λ_1
500	0.446 (0.459)	1.200 (0.902)	3.927 (0.855)
1000	0.346 (0.154)	1.020 (0.549)	3.940 (0.514)
2000	0.321 (0.102)	0.949 (0.343)	3.922 (0.361)
n	λ_2	μ	
500	17.380 (3.811)	0.620 (0.061)	
1000	17.850 (2.073)	0.615 (0.043)	
2000	18.016 (1.483)	0.614 (0.030)	

$\mu^0 = 1.7$

n	q_1	q_2	λ_1
500	0.442 (0.458)	1.196 (0.894)	3.945 (0.858)
1000	0.342 (0.152)	1.018 (0.547)	3.954 (0.512)
2000	0.318 (0.100)	0.945 (0.342)	3.935 (0.360)
n	λ_2	μ	
500	17.417 (3.829)	0.621 (0.061)	
1000	17.897 (2.068)	0.616 (0.043)	
2000	18.051 (1.480)	0.615 (0.030)	

Table 2.4: Mean of estimates of 100 data sets, with corresponding standard deviation between brackets, for $\Delta = 0.05$ and $\mu^0 = 0.1$ in the upper part and $\mu^0 = 1.7$ in the bottom part. True parameter values: $q_1 = 0.3, q_2 = 0.9, \lambda_1 = 4, \lambda_2 = 18$, $\mu = 0.6$.

To investigate the accuracy of the algorithm presented in this section, and for a proper comparison with the results in Section 2.4, we extended the simulation study of Section 2.4 by using the same data vectors and corresponding initial values, but adding the parameter update $\tilde{\mu}$ in (2.42) to the algorithm. Various initial values μ^0 for μ were taken, of which for two values, $\mu^0 = 0.1$ and $\mu^0 = 1.7$, the results are shown. Table 2.4 shows for each sample size (rows) and for each parameter (columns), the mean values of the 100 estimates together with the corresponding standard deviation between brackets, with $\mu^0 = 0.1$ and $\mu^0 = 1.7$ respectively. Our experiments show that the choice of μ^0 has a minor effect on the estimates, as each of the μ^0 values leads to very similar results. We see in Table 2.4 that the means of the estimates for μ lie close to the true parameter value and the corresponding standard deviations are small. This can be explained by the fact that the parameter μ is not affected by the modulation. We also see that the standard deviations decrease as n increases, which is also visible in Figure 2.5. Here the histograms of the 100 estimates for μ are shown with increasing value of n, and $\mu^0 = 0.1$. The histograms show that the estimates for μ are concentrated around the true parameter value, and they have a bell-shape, indicating the distribution of the estimator being approximately normal. Finally, the values in Tables 2.2 and 2.4 illustrate that the estimates and corresponding standard deviations of the other parameters are barely influenced by the estimation of μ.

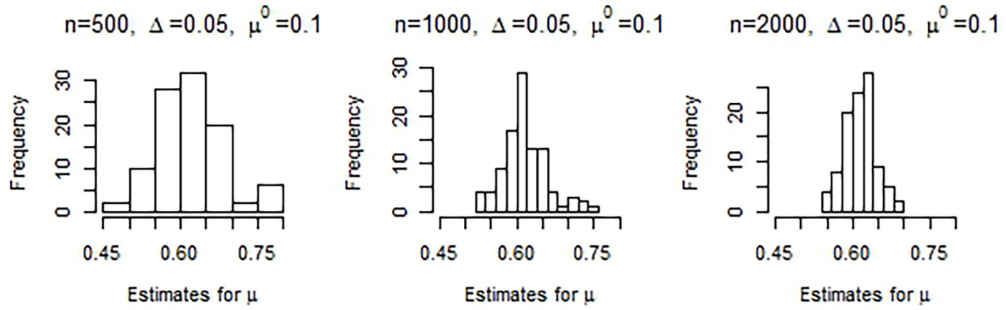

Figure 2.5: Histograms of the obtained estimates for μ, with n increasing from left to right, and $\mu^0 = 0.1$.

2.6 Discussion

In this chapter we developed an algorithm for finding estimates of the parameters of an MMIS process. The proposed algorithm numerically approximates the maximum likelihood estimates of the parameters based on observations of the population size at equidistant time points. The algorithm is an iterative EM-type algorithm, in the spirit of the one that Okamura et al. [50] developed. It is stressed that they did not consider departures but rather assumed discrete-time observations of the cumulative arrival process, and therefore some major adjustments to the steps in their approach were required to obtain our estimation algorithm.

We have investigated the accuracy of the proposed algorithm by means of an extensive simulation study. The results showed that the estimates are concentrated around the true parameter values and become more accurate as the sample size increases. In addition, the results indicated that for sufficiently large n the distributions of the estimators become approximately normal. Furthermore, the estimates got more accurate when the background process becomes slower, as it is easier for the algorithm to detect the different states. However, to retain a high accuracy, the sample size n, and hence $T = n\Delta$, must be large such that the background process jumps often enough during the total observation time T. Moreover, for larger d, n must also be larger to retain a similar accuracy.

The run-time of the algorithm depends on various parameters. In the first place, the run-time of a single iteration in the algorithm is linear in n. However, as n increases the algorithm is likely to converge more quickly, implying that the number of iterations required will decrease in n. In addition, the run-time of the algorithm increases in d. The run-time of one iteration in the algorithm is mainly determined by step 3 in Section 2.3.6, and from Section 2.3.5 we know that the computational effort of this step increases in d. However, as we mentioned before, Markov-modulation is typically used to model situations for which d is small (typically 2). We finally mention that the number of iterations needed for convergence of the algorithm, and hence the run-time, tends to be large if the length of the observation intervals, Δ, is large, or if the initial parameter values are far away from the true parameter values.

There are many interesting directions for future research based on our results. We note that the estimation algorithm that we developed is built on the assumption that we know the number of states d. The choice of the dimension d from the data is a model selection problem which is outside the scope of this chapter, but could be explored in a follow-up project. Another research theme could relate to generalizing the sojourn time distribution, where non-parametric estimation could be explored. One could also consider more general inter-arrival times, since for some applications exponential inter-arrival times may not be a suitable fit.

3. Quasi birth-death processes

Continuous-time quasi birth-death (QBD) processes can informally be seen as birth-death processes of which the parameters are modulated by an external continuous-time Markov chain. The aim is to numerically approximate the time-dependent distribution of the resulting bivariate Markov process in an accurate and efficient way. An approach based on the Erlangization principle is proposed and formally justified. Its performance is investigated and compared with two existing approaches: one based on numerical evaluation of the matrix exponential underlying the QBD process, and one based on the uniformization technique. It is shown that in many settings the approach based on Erlangization is faster than the other approaches, while still being highly accurate. We demonstrate the use of the developed technique in the context of the evaluation of the likelihood pertaining to a time series, which can then be optimized over its parameters to obtain the maximum likelihood estimator. More specifically, through a series of examples with simulated and real-life data, we show how it can be deployed in model selection problems that involve the choice between a QBD and its non-modulated counterpart.

3.1 Introduction

Birth-death (BD) processes are continuous-time Markov processes where transitions can only increase or decrease the state by one—usually referred to as births and deaths, respectively. These well-known processes are widely used and have applications in many areas such as biology, epidemiology and operations research. In some real-life systems, however, it is likely that there is a higher variability in the birth- and/or the death rates than modelled by a conventional BD process. Observe for example the data in Figure 3.1, displaying the annual counts of the female population of the whooping crane (see [66] for the original data, and [24] for the female counts). There are some fluctuations visible in the evolution of the population size, which could be indicative of a higher variability in some, or all, model parameters. One wonders whether specific generalizations of the BD process could be more suitable for this data. The major aim of this chapter is to develop methodologies that can be used to rigorously compare the fit of a conventional BD process with more general alternatives.

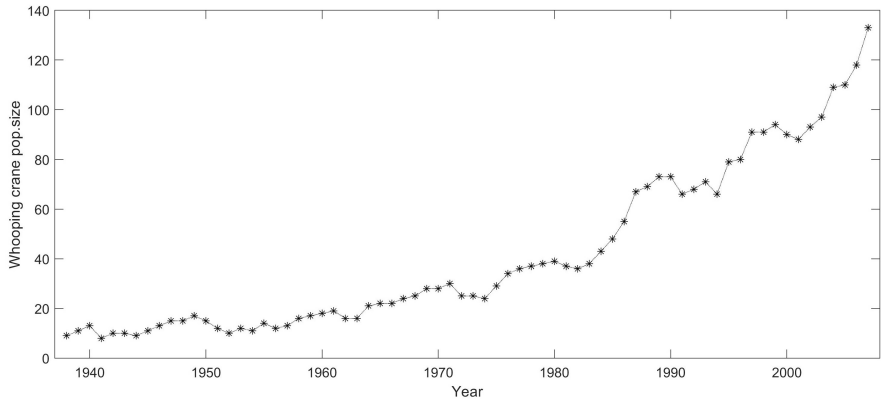

Figure 3.1: Yearly population count of female whooping cranes arriving in Texas each autumn.

An example of a more general alternative to the conventional BD process is the *quasi birth-death* (QBD) process. The population process, called the level process, in a QBD process is given by a BD process of which the transition rates are modulated by a continuous-time Markov chain, called the phase process. This means that the transition rates of the QBD process switch between multiple distinct values at the jump times of the phase process. Together, the level and the phase process form a bivariate Markov process. In an even more general QBD process, the number of states of the phase process can depend on the current value of the level process. This leads to a so-called level-dependent QBD process, which is the process that we consider in this chapter. Over the years, various properties of level-dependent QBD processes have been studied. We refer to e.g. [16] for calculations concerning the equilibrium distribution, [57] for the computation of certain matrices that play an important role in the QBD context, and [43] for a characterization of the process' running maximum.

In the above whooping crane example, one would like to statistically compare the scenario of the data stemming from a conventional BD process with that of the data stemming from the more general QBD process. In order to do so, a prerequisite is that we have a methodology to compute, for both models, the likelihood of our dataset. This, in turn, requires techniques for the evaluation of the time-dependent probabilities corresponding to BD and QBD processes. In this chapter we investigate different approaches to compute the time-dependent probabilities of the joint Markov process of level and phase in the level-dependent QBD process. In particular, we propose, justify and test an approach based on the so-called *Erlangization* principle, which we compare with existing alternatives. Then we point out through a series of experiments, including the whooping crane example, how such techniques can be used in determining whether a BD process or a QBD process yields the better fit.

In order to numerically evaluate probabilities pertaining to BD and QBD processes, various methods have been developed. For all practical purposes, it is natural to let the underlying Markov chain live on a finite state space. A commonly applied approach to compute the time-dependent distribution boils down to computing the matrix exponential

of the transition rate matrix, say Q, of the corresponding Markov chain (of which the states, in the QBD case, encode all level/phase combinations). More precisely, the (i, j)-th entry of e^{Qt} provides us with the probability of being in state j at time t given that the initial state was i, where in the QBD context, i and j correspond to specific phase/level combinations. It is known, however, that the computation of matrix exponentials may involve various numerical complications; see e.g. the survey [46]. Various novel, more sophisticated approaches are being developed [2], but, citing [46], 'none are completely satisfactory'. Alternatively, one could solve the linear system of differential equations resulting from the Kolmogorov equations. As argued in e.g. [59], this method has various intrinsic problems as well. Most notably, if the underlying system is large, the Q matrix is ill-conditioned, or the differential equations are stiff, the evaluation can be slow and/or inaccurate.

Owing to the special structure of the transition rate matrix (i.e., the Q-matrix having non-negative off-diagonal entries, row sums equal to 0), another approach is possible. In the *uniformization* technique the continuous-time Markov chain is converted to a discrete-time Markov chain (say with transition matrix P) of which the jump times correspond to a Poisson process with a constant rate (say σ). Here P and σ are chosen in such a way that the newly defined process and the original continuous-time Markov chain are statistically identical, i.e. all distributional properties are equivalent. The distribution of the continuous-time Markov chain at time t can thus be obtained by weighing matrices P^k by the probabilities that the Poisson process has jumped k times in $[0, t]$, and summing these over k ($k = 0, 1, \ldots$). This method performs well in many cases, but it has disadvantages as well. Evidently, in numerical computations the above summation has to be truncated at some finite threshold, where the issue is to choose this threshold high enough to make sure that the error made is negligible. In addition, to compute all k-step transition matrices P^k, the corresponding matrix multiplications need to be executed, which may make the procedure prohibitively slow. Uniformization was introduced in the 1950s in [35]; see also [28, 29, 45] for other seminal contributions; an extensive discussion on its pros and cons can be found in [68].

In this chapter we discuss an alternative approach, based on the Erlangization principle, which has previously been explored (in other contexts) in e.g. [7, 58, 43]. It uses the fact that, although the computation of the distribution of the state of the Markov chain at a deterministic time is challenging, its counterpart at an exponentially distributed epoch just requires solving a system of linear equations. A second observation is that the sum of k independent exponentially distributed random variables with mean t/k—corresponding to an Erlang distribution with scale parameter k and shape parameter k/t—converges to the deterministic number t as k grows large. Combining these two properties, the idea is to evaluate the transition probabilities at an exponentially distributed epoch with parameter k/t, and to raise the resulting matrix to the power k. It is tempting to believe that our deterministic-time transition probabilities are accurately approximated by this procedure as long as k is chosen large enough. This approach has the inherent advantage that the number of matrix multiplications is limited: if k is a power of two, it suffices to square the exponential-time transition matrix $\log_2 k$ times. Importantly, we can prove the theoretical correctness of the approach, in that we show that it becomes increasingly

precise as $k \to \infty$, with an argumentation that relies on large-deviations theory. By means of a series of numerical examples, we also show that this approach is in many settings computationally faster than the approaches based on the matrix exponential and uniformization, without compromising the accuracy.

Going back to the whooping crane data from Figure 3.1, an interesting question remains if a QBD process indeed provides a better fit to the data than a conventional BD process, as one might suspect from the graph. In the last section of this chapter we investigate this type of model selection problems, both with simulated and real-life data. By the techniques discussed in this chapter, we can compute the likelihood pertaining to a time series, thus enabling the evaluation of maximum likelihood estimates. In this respect, note that all three approaches (i.e., matrix exponential, uniformization, Erlangization) can be applied in the QBD as well as the BD setting. As the class of QBD processes contains the class of BD processes, evidently the former by definition leads to a better fit, but this comes at the price of additional parameters. To 'fairly' compare the two models, taking into account the corresponding numbers of parameters, we perform the model selection relying on the celebrated Akaike information criterion (AIC).

The remainder of this chapter is organized as follows. The level-dependent QBD process and its corresponding time-dependent distribution are defined in Section 3.2. Section 3.3 shows how the transition probabilities at an exponentially distributed epoch can be computed by solving a system of linear equations. The findings of Section 3.3 are then used in Section 3.4 to motivate the Erlangization approach; in addition the theoretical correctness of this approach is established. Section 3.5 experimentally investigates the performance of the three approaches discussed above. Section 3.6 discusses the model selection problem of choosing between BD processes and QBD processes, using examples with simulated as well as real-life data, with all likelihood computations relying on Erlangization. We conclude the chapter, in Section 3.7, with a brief discussion.

3.2 Model and preliminaries

In this section we introduce the class of QBD processes that will be considered in this chapter. Next, we define the object of our study, viz. the time-dependent distribution of the corresponding bivariate Markov process, and briefly discuss established approaches to numerically evaluate it.

3.2.1 Model

A QBD process is a bivariate process comprising *levels* and *phases*. The level process, in the sequel denoted by $\{M_t\}_{t\geqslant 0}$, attains values in $\{0, 1, \ldots, C\}$ for some $C \in \mathbb{N}$. The phase process is denoted by $\{X_t\}_{t\geqslant 0}$; when the level M_t equals m, the phase X_t attains values in $\{1, \ldots, d_m\}$, for some $d_m \in \mathbb{N}$. In many applications the number of phases is uniform in the level, or, more concretely, $d_m = d \in \mathbb{N}$ for all $m \in \{0, \ldots, C\}$. The birth-death

nature of the process is reflected by the fact that at any transition the level can increase or decrease by at most 1.

We provide a more precise description of the model $\{M_t, X_t\}_{t\geqslant 0}$ by formally defining the corresponding transition rates.

- In the first place, $Q^{(m)}$, for $m \in \{0, 1, \ldots, C\}$, is a transition rate matrix of dimension $d_m \times d_m$ that corresponds to a continuous-time Markov chain living on the state space $\{1, \ldots, d_m\}$. Its elements are denoted by $q_{ij}^{(m)}$; they are non-negative for $i \neq j$ and in addition the row sums are zero. Whenever $M_t = m$, a jump from phase i to phase j *that leaves the level unchanged* occurs with rate $q_{ij}^{(m)}$, for $i \neq j$. In addition, we define the total rate out of phase i (while the level remains at m),

$$q_i^{(m)} := -q_{ii}^{(m)} = \sum_{j\neq i} q_{ij}^{(m)};$$

  here the sum on the right hand side should be understood to be over all $j \in \{1, \ldots, d_m\}$ such that $j \neq i$.

- In the second place, there are transitions in which the level goes up by 1, while at the same time the phase potentially changes as well. For $m \in \{0, 1, \ldots, C-1\}$, the matrix $\Lambda^{(m)}$ has dimension $d_m \times d_{m+1}$. Its (i, j)-th element contains the rate $\lambda_{ij}^{(m)} \geqslant 0$ at which the level increases by 1 while simultaneously the phase jumps from i to j; note that $i = j$ is allowed (under the proviso that $i \leqslant \min\{d_m, d_{m+1}\}$). Throughout this chapter we write

$$\lambda_i^{(m)} := \sum_{j=1}^{d_{m+1}} \lambda_{ij}^{(m)},$$

  to denote the total rate corresponding to an increase in level from phase i, with $i \in \{1, \ldots, d_m\}$.

- Finally, there are transitions in which the level goes down by 1, again potentially simultaneously with a phase change. The (i, j)-th element of the matrix $\mathcal{M}^{(m)}$, which has dimension $d_m \times d_{m-1}$ for $m \in \{1, 2, \ldots, C\}$, contains the rate $\mu_{ij}^{(m)} \geqslant 0$ at which the level decreases by 1 while the phase jumps from i to j; again, $i = j$ is allowed (if $i \leqslant \min\{d_{m-1}, d_m\}$). We compactly write for the total rate of a decrease in level from phase i, with $i \in \{1, \ldots, d_m\}$,

$$\mu_i^{(m)} := \sum_{j=1}^{d_{m-1}} \mu_{ij}^{(m)}.$$

In this work we assume that the matrices $Q^{(m)}$, $\Lambda^{(m)}$, and $\mathcal{M}^{(m)}$ are such that the joint Markov process $\{M_t, X_t\}_{t\geq 0}$ is irreducible, implying that, with positive probability any

level/phase pair can be reached from any other level/phase pair in any amount of time. The number of states of this process is $D := \sum_{m=0}^{C} d_m$. We let Q be the $D \times D$ transition rate matrix of $\{M_t, X_t\}_{t \geq 0}$, that is,

$$Q := \begin{pmatrix} \bar{Q}^{(0)} & \Lambda^{(0)} & 0 & \cdots & 0 & 0 \\ \mathcal{M}^{(1)} & \bar{Q}^{(1)} & \Lambda^{(1)} & \cdots & 0 & 0 \\ 0 & \mathcal{M}^{(2)} & \bar{Q}^{(2)} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & \bar{Q}^{(C-1)} & \Lambda^{(C-1)} \\ 0 & 0 & 0 & \cdots & \mathcal{M}^{(C)} & \bar{Q}^{(C)} \end{pmatrix},$$

where $\bar{Q}^{(m)}$ is defined as $Q^{(m)}$ with the diagonal entries adapted such that the row sums of Q are zero. More precisely, the definition of $\bar{Q}^{(m)}$ entails that the diagonal of Q consists of entries of the form $-\sigma_i^{(m)}$, where (for $m \in \{0, 1, \ldots, C\}$ and $i \in \{1, \ldots, d_m\}$)

$$\sigma_i^{(m)} := q_i^{(m)} + \lambda_i^{(m)} 1_{\{m<C\}} + \mu_i^{(m)} 1_{\{m>0\}}. \tag{3.1}$$

These rates $\sigma_i^{(m)}$ are to be interpreted as the 'total flux' when the level is m and the phase is i. For later reference we define the largest entry among these fluxes by

$$\sigma := \max_{m \in \{0,1,\ldots,C\}} \left(\max_{i \in \{1,\ldots,d_m\}} \sigma_i^{(m)} \right). \tag{3.2}$$

We finally introduce the $D \times D$ matrix P_t that describes the process' time-dependent distribution. It contains probabilities of the type

$$p_{ij}(m, m'; t) := \mathbb{P}(M_t = m', X_t = j \,|\, M_0 = m, X_0 = i), \tag{3.3}$$

with the states ordered in the same way as is done in Q. The remainder of this section is devoted to describing two often used methods to numerically evaluate P_t, with which we compare our method in Section 3.5.

3.2.2 Time-dependent probabilities: matrix exponential

It is commonly known that P_t, as given in (3.3), can be expressed as a matrix exponential, i.e., $P_t = e^{Qt}$. As argued extensively in [46], the numerical evaluation of such matrix exponentials is a delicate issue. In numerical computing environments various types of algorithms have been implemented. MATLAB's implementation `expm`$(\cdot)$ is based on the algorithm developed in [33], and is claimed to be highly accurate; see also the further refinements in [2].

Approximation 3.1 (Matrix exponential)**.** P_t is approximated by

$$P_t^{(\mathrm{m})} := \texttt{expm}(Qt), \tag{3.4}$$

based on MATLAB's implementation $\texttt{expm}(\cdot)$.

3.2.3 Time-dependent probabilities: uniformization

An alternative existing approach to obtain time-dependent probabilities relies on uniformization. The main idea is to convert the continuous-time Markov chain to a discrete-time Markov chain of which the jump times follow a Poisson process with a constant rate. For the QBD process we let this uniform rate be σ, as defined in (3.2). Define, with self-evident notation,

$$\mathscr{P}_{(m,i),(m',j)} := \begin{cases} \sigma^{-1} Q_{(m,i),(m',j)} & \text{if } (m,i) \neq (m',j), \\ 1 - \sigma^{-1} \sum_{(m',j)\neq(m,i)} Q_{(m,i),(m',j)} & \text{if } (m,i) = (m',j), \end{cases}$$

or, equivalently, $Q = \sigma\mathscr{P} - \sigma I$. To Observe that by definition of σ all these entries are in $[0,1]$; in fact, $\mathscr{P}$ is a transition probability matrix of a discrete-time Markov chain. Sampling the number of jumps in $(0,t]$ of this discrete-time Markov chain according to a Poisson distribution with parameter σt, we find that

$$P_t = e^{Qt} = e^{(\sigma\mathscr{P}-\sigma I)t} = \sum_{k=0}^{\infty} e^{-\sigma t} \frac{(\sigma t)^k}{k!} \mathscr{P}^k,$$

The following approximation is based on this representation.

Approximation 3.2 (Uniformization)**.** For a given $\ell \in \mathbb{N}$, P_t is approximated by

$$P_t^{(\mathrm{u},\ell)} := \sum_{k=0}^{\ell} e^{-\sigma t} \frac{(\sigma t)^k}{k!} \mathscr{P}^k. \tag{3.5}$$

A question is: how to select a value of ℓ to make sure that the error made is below some allowable threshold $\varepsilon > 0$? To this end, realize that, trivially, as $\ell \to \infty$,

$$0 \leqslant p_{ij}(m,m';t) - p_{ij}^{(\mathrm{u},\ell)}(m,m';t) \leqslant \mathbb{P}(\mathrm{Pois}(\sigma t) \geqslant \ell + 1) \to 0,$$

where $\mathrm{Pois}(\sigma t)$ denotes a Poisson random variable with mean σt. This bound entails that

one could use for example the Chernoff bound to find the ℓ for which $\mathbb{P}(\text{Pois}(\sigma t) \geqslant \ell + 1) < \varepsilon$:

$$\begin{aligned}\mathbb{P}(\text{Pois}(\sigma t) \geqslant \ell + 1) &\leqslant \inf_{\theta>0} e^{-\theta(\ell+1)}\, \mathbb{E}\, e^{\theta \,\text{Pois}(\sigma t)} \\ &= \inf_{\theta>0} e^{-\theta(\ell+1)}\, e^{\sigma t(e^{\theta}-1)} = \left(\frac{\sigma t}{\ell+1}\right)^{\ell+1} e^{\ell+1-\sigma t}; \end{aligned} \tag{3.6}$$

equating the right-hand side to ε yields an ℓ with the desired property.

Note that an important advantage of uniformization is its implementational simplicity: the matrix $\mathscr{P}$ is trivially computed from Q, and it is straightforward to evaluate its powers. The main disadvantage of uniformization is that *many* matrix multiplications are needed, as the approximation uses *all* matrices $\mathscr{P}^k$ for $k = \{0, 1, \ldots, \ell\}$; particularly when σ is relatively large, implying that ℓ has to be chosen large as well, the procedure may become rather time consuming. To remedy this disadvantage of uniformization, we pursue an alternative approach, based on the concept of *Erlangization*. This approach combines two ideas: (i) if the time horizon is exponentially distributed rather than deterministic, then the corresponding transition probability follows simply by solving a linear system of equations, and (ii) one can approximate a deterministic number by a sum of a large number of independent exponentially distributed random variables with an appropriately chosen parameter. Section 3.3 first elaborates on property (i). Then, in Section 3.4, it is pointed out how, based on these two properties, P_t can be efficiently and accurately approximated. In Section 3.5 we numerically compare the performance of Erlangization with the matrix exponential approach (3.4) and uniformization (3.5).

3.3 Time-dependent probabilities at exponential epochs

The main goal of this section is to show that the evaluation of the distribution of $\{M_t, X_t\}$ at an exponentially distributed epoch essentially reduces to solving a linear system of equations. Let T_η be an exponentially distributed random variable with mean η^{-1} (with $\eta > 0$), independent of $\{M_t, X_t\}_{t\geqslant 0}$. We define

$$\pi_{ij}(m, m'; \eta) := \mathbb{P}(M_{T_\eta} = m', X_{T_\eta} = j \,|\, M_0 = m, X_0 = i).$$

We now point out how to compute these probabilities $\pi_{ij}(m, m'; \eta)$, with $m, m' \in \{0, 1, \ldots, C\}$, $i \in \{1, \ldots, d_m\}$, and $j \in \{1, \ldots, d_{m'}\}$. Recall the definition of $\sigma_i^{(m)}$ in (3.1). The standard

'Markovian reasoning' yields

$$\begin{aligned}
&\pi_{ij}(m,m';\eta)\\
&\quad= \sum_{i'=1,i'\neq i}^{d_m} \frac{q_{ii'}^{(m)}}{(\sigma_i^{(m)}+\eta)}\,\pi_{i'j}(m,m';\eta) + \sum_{i'=1}^{d_{m+1}} \frac{\lambda_{ii'}^{(m)}}{(\sigma_i^{(m)}+\eta)}\,\pi_{i'j}(m+1,m';\eta)\,1_{\{m<C\}}\\
&\qquad+ \sum_{i'=1}^{d_{m-1}} \frac{\mu_{ii'}^{(m)}}{(\sigma_i^{(m)}+\eta)}\,\pi_{i'j}(m-1,m';\eta)\,1_{\{m>0\}} + \frac{\eta}{(\sigma_i^{(m)}+\eta)}\,1_{\{m=m',i=j\}}.
\end{aligned}$$

Multiplying both sides of the equation with $\sigma_i^{(m)}+\eta$ results in

$$\begin{aligned}
&(\sigma_i^{(m)}+\eta)\,\pi_{ij}(m,m';\eta)\\
&\quad= \sum_{i'=1,i'\neq i}^{d_m} q_{ii'}^{(m)}\,\pi_{i'j}(m,m';\eta) + \sum_{i'=1}^{d_{m+1}} \lambda_{ii'}^{(m)}\,\pi_{i'j}(m+1,m';\eta)\,1_{\{m<C\}}\\
&\qquad+ \sum_{i'=1}^{d_{m-1}} \mu_{ii'}^{(m)}\,\pi_{i'j}(m-1,m';\eta)\,1_{\{m>0\}} + \eta\,1_{\{m=m',i=j\}}.
\end{aligned}$$

The sum of the coefficients on the right equals $\sigma_i^{(m)}+\eta$, making this system of linear equations strictly diagonally dominant, and therefore non-singular [34, Thm 6.1.10]. As a consequence, the system can be numerically solved in $\pi_{ij}(m,m';\eta)$ through various efficient evaluation techniques, such as the iterative Jacobi and Gauss-Seidel methods [9, Section VIII.6].

The above linear system can be written in a compact matrix form. Define the $d_m \times d_{m'}$ matrix $\Pi_\eta(m,m')$ as the matrix whose (i,j)-th entry is $\pi_{ij}(m,m';\eta)$. In addition, let $\Sigma^{(m)} := \mathrm{diag}\{\sigma_1^{(m)},\ldots,\sigma_{d_m}^{(m)}\}$ and $\check{Q}^{(m)} := \mathrm{diag}\{q_1^{(m)},\ldots q_{d_m}^{(m)}\}$; the matrix $I^{(m)}$ is an identity matrix of dimension d_m. We thus obtain

$$\begin{aligned}
(\Sigma^{(m)}+\eta I^{(m)})\Pi_\eta(m,m') = (Q^{(m)}+\check{Q}^{(m)})\Pi_\eta(m,m') + \Lambda^{(m)}\Pi_\eta(m+1,m')1_{\{m<C\}}\\
+ \mathcal{M}^{(m)}\Pi_\eta(m-1,m')1_{\{m>0\}} + \eta I^{(m)}1_{\{m=m'\}}.
\end{aligned}$$

Observe that in the above linear equations the level at time T_η is constant (namely, m'). We can therefore solve the matrices $\Pi_\eta(m,m')$ (with $m=0,1,\ldots,C$) for each m' separately; notice that for a given m' this concerns $d_{m'}\,D$ equations in equally many unknowns.

We define Π_η as a $D\times D$ matrix, which is a block matrix of which the components

are the matrices $\Pi_\eta(m, m')$:

$$\Pi_\eta := \begin{pmatrix} \Pi_\eta(0,0) & \Pi_\eta(0,1) & \Pi_\eta(0,2) & \cdots & \Pi_\eta(0,C) \\ \Pi_\eta(1,0) & \Pi_\eta(1,1) & \Pi_\eta(1,2) & \cdots & \Pi_\eta(1,C) \\ \Pi_\eta(2,0) & \Pi_\eta(2,1) & \Pi_\eta(2,2) & \cdots & \Pi_\eta(2,C) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Pi_\eta(C,0) & \Pi_\eta(C,1) & \Pi_\eta(C,2) & \cdots & \Pi_\eta(C,C) \end{pmatrix}. \tag{3.7}$$

This matrix Π_η will appear in the approximation of P_t based on Erlangization, introduced in the next section.

3.4 Erlangization

In this section, we discuss the approach based on Erlangization to approximate P_t. We first introduce the approximation and then provide the theoretical correctness of this approach. Let $S_{\ell,t}$ be an Erlang-distributed random variable with rate parameter ℓ/t and shape parameter ℓ. Let $P_t^{(\mathrm{e},\ell)}$ be a $D \times D$ matrix with entries

$$p_{ij}^{(\mathrm{e},\ell)}(m, m'; t) := \mathbb{P}(M_{S_{\ell,t}} = m', X_{S_{\ell,t}} = j \mid M_0 = m, X_0 = i).$$

It is clear that $P_t^{(\mathrm{e},\ell)} = (\Pi_{\ell/t})^\ell$, with Π_η as defined in (3.7), owing to the fact that an Erlang random variable with rate parameter μ and shape parameter k can be written as the sum of k independent and identically distributed exponential random variables with rate μ. We propose the following approximation.

Approximation 3.3 (Erlangization)**.** For a given $\ell \in \mathbb{N}$, P_t is approximated by,

$$P_t^{(\mathrm{e},\ell)} = (\Pi_{\ell/t})^\ell. \tag{3.8}$$

As we will argue below, $P_t^{(\mathrm{e},\ell)}$ converges to P_t as $\ell \to \infty$. The above idea is usually referred to as 'Erlangization': the time $t \geqslant 0$ is approximated by the Erlang time $S_{\ell,t}$. This distribution has mean t and variance t^2/ℓ, so that the corresponding coefficient of variation converges to 0 as $\ell \to \infty$.

Our goal is to assess how much $p_{ij}(m, m'; t)$ differs from $p_{ij}^{(\mathrm{e},\ell)}(m, m'; t)$. The resulting bounds are then used to show that this difference vanishes as ℓ grows large. We start by

establishing an upper bound. For any given $\varepsilon \in (0, t)$,

$$\begin{aligned}
&p_{ij}^{(\mathrm{e},\ell)}(m, m'; t) \\
&\quad = \mathbb{P}(M_{S_{\ell,t}} = m', X_{S_{\ell,t}} = j \,\big|\, |S_{\ell,t} - t| \leqslant \varepsilon, M_0 = m, X_0 = i)\, \mathbb{P}(|S_{\ell,t} - t| \leqslant \varepsilon) \\
&\qquad + \mathbb{P}(M_{S_{\ell,t}} = m', X_{S_{\ell,t}} = j \,\big|\, |S_{\ell,t} - t| > \varepsilon, M_0 = m, X_0 = i)\, \mathbb{P}(|S_{\ell,t} - t| > \varepsilon) \\
&\quad \leqslant \mathbb{P}(M_{S_{\ell,t}} = m', X_{S_{\ell,t}} = j \,|\, |S_{\ell,t} - t| \leqslant \varepsilon, M_0 = m, X_0 = i) + \mathbb{P}(|S_{\ell,t} - t| > \varepsilon).
\end{aligned}$$

Note that $\mathbb{P}(M_{S_{\ell,t}} = m', X_{S_{\ell,t}} = j \,|\, |S_{\ell,t} - t| \leqslant \varepsilon, M_0 = m, X_0 = i)$ is equal to the transition probability $p_{ij}(m, m'; S_{\ell,t})$ additionally imposing the condition that $|S_{\ell,t} - t| \leqslant \varepsilon$. The difference between this probability and $p_{ij}(m, m'; t)$ can thus be at most ε times the maximum slope of $p_{ij}(m, m'; s)$ for s in $[t - \varepsilon, t + \varepsilon]$. Hence

$$p_{ij}^{(\mathrm{e},\ell)}(m, m'; t) \leqslant p_{ij}(m, m'; t) + \varepsilon \left(\sup_{s \in [t-\varepsilon, t+\varepsilon]} \left| \frac{\mathrm{d}}{\mathrm{d}s} p_{ij}(m, m'; s) \right| \right) + \mathbb{P}(|S_{\ell,t} - t| > \varepsilon).$$

Recall that Q is the transition rate matrix of the D-dimensional continuous-time Markov process $\{M_t, X_t\}_{t \geqslant 0}$ and $\sigma := \max_{m,i} \sigma_i^{(m)}$. Then, using the Kolmogorov equations in combination with the triangle inequality, uniformly in $s \geqslant 0$,

$$\left| \frac{\mathrm{d}}{\mathrm{d}s} p_{ij}(m, m'; s) \right| \leqslant \sum_{m'', j'} p_{ij'}(m, m''; s) \, \left| Q_{(m'', j'), (m', j)} \right| \leqslant \sum_{m'', j'} p_{ij'}(m, m''; s)\, \sigma = \sigma.$$

We proceed by finding an upper bound on $\mathbb{P}(|S_{\ell,t} - t| > \varepsilon)$. Noting that $S_{\ell,t}$ can be written as ℓ^{-1} times an Erlang random variable $\bar{S}_{\ell,t}$ with rate parameter $1/t$ and shape parameter ℓ,

$$\mathbb{P}(|S_{\ell,t} - t| > \varepsilon) = \mathbb{P}(|\ell^{-1} \bar{S}_{\ell,t} - t| > \varepsilon) = \mathbb{P}(\ell^{-1} \bar{S}_{\ell,t} - t < -\varepsilon) + \mathbb{P}(\ell^{-1} \bar{S}_{\ell,t} - t > \varepsilon). \quad (3.9)$$

We can majorize both probabilities on the right-hand side by using the Chernoff bound. Starting with $\mathbb{P}(\ell^{-1} \bar{S}_{\ell,t} - t > \varepsilon)$, we have

$$\mathbb{P}(\ell^{-1} \bar{S}_{\ell,t} - t > \varepsilon) = \mathbb{P}(\bar{S}_{\ell,t} > \ell(\varepsilon + t)) \leqslant \inf_{\theta > 0} e^{-\theta\, \ell(\varepsilon + t)} \mathbb{E}\, e^{\theta \bar{S}_{\ell,t}}.$$

Using the moment generating function of the Erlang distribution, we find that

$$e^{-\theta\, \ell(\varepsilon + t)} \mathbb{E}\, e^{\theta \bar{S}_{\ell,t}} = \left(\frac{e^{-\theta(\varepsilon + t)}}{1 - t\theta} \right)^{\ell},$$

implying that

$$\mathbb{P}(\ell^{-1}\bar{S}_{\ell,t} - t > \varepsilon) \leqslant \inf_{\theta>0} \left(\frac{e^{-\theta(\varepsilon+t)}}{1-t\theta} \right)^{\ell} = \left(\inf_{\theta>0} \frac{e^{-\theta(\varepsilon+t)}}{1-t\theta} \right)^{\ell} = e^{-\ell\varepsilon/t} \left(1 + \frac{\varepsilon}{t}\right)^{\ell}.$$

In a similar way we can majorize $\mathbb{P}(\ell^{-1}\bar{S}_{\ell,t} - t < -\varepsilon)$:

$$\mathbb{P}(\ell^{-1}\bar{S}_{\ell,t} - t < -\varepsilon) \leqslant e^{\ell\varepsilon/t} \left(1 - \frac{\varepsilon}{t}\right)^{\ell}.$$

Combining these upper bounds with equation (3.9), we conclude

$$\mathbb{P}(|S_{\ell,t} - t| > \varepsilon) \leqslant e^{\ell\varepsilon/t} \left(1 - \frac{\varepsilon}{t}\right)^{\ell} + e^{-\ell\varepsilon/t} \left(1 + \frac{\varepsilon}{t}\right)^{\ell} =: \Psi_{\ell,t}(\varepsilon). \tag{3.10}$$

We thus find, uniformly in $\varepsilon \in (0, t)$,

$$p_{ij}^{(\mathrm{e},\ell)}(m, m'; t) \leqslant p_{ij}(m, m'; t) + \varepsilon \cdot \sigma + \Psi_{\ell,t}(\varepsilon).$$

Now take $\varepsilon = \ell^{-\alpha}$ with $\alpha > 0$. Using elementary Taylor expansions, it can be shown that $\Psi_{\ell,t}(\varepsilon)$ behaves as $\exp(-\ell^{1-2\alpha}/t^2)$, which converges to 0 as $\ell \to \infty$ for all $\alpha < 1/2$. To see this, first note that

$$e^{\ell\varepsilon/t} \left(1 - \frac{\varepsilon}{t}\right)^{\ell} = \exp\left(\frac{\ell}{t}\varepsilon + \ell \log\left(1 - \frac{\varepsilon}{t}\right) \right). \tag{3.11}$$

Now consider the exponent in the right-hand side of (3.11). Plugging in $\varepsilon = \ell^{-\alpha}$ and using Taylor expansions, one indeed obtains

$$\frac{1}{t}\ell^{1-\alpha} + \ell \log\left(1 - \frac{1}{t}\ell^{-\alpha}\right) = -\frac{1}{t^2}\ell^{1-2\alpha} + o(\ell^{1-3\alpha}).$$

A similar analysis can be performed for the other term in the definition of $\Psi_{\ell,t}(\varepsilon)$. We conclude that, for all $\alpha < 1/2$, $\Psi_{\ell,t}(\ell^{-\alpha})$ converges to 0 as $\ell \to \infty$. Upon combining the above, and picking $\alpha = \frac{1}{3}$, the desired upper bound follows:

$$\limsup_{\ell\to\infty} p_{ij}^{(\mathrm{e},\ell)}(m, m'; t) \leqslant \limsup_{\ell\to\infty} p_{ij}(m, m'; t) + \ell^{-1/3} \cdot \sigma + \Psi_{\ell,t}(\ell^{-1/3}) = p_{ij}(m, m'; t).$$

We proceed by deriving a lower bound, which is established using elements that resemble those used in the upper bound. It is based on the inequality

$$\begin{aligned} & p_{ij}^{(\mathrm{e},\ell)}(m,m';t) \\ & \quad \geqslant \mathbb{P}(M_{S_{\ell,t}} = m', X_{S_{\ell,t}} = j \,|\, M_0 = m, X_0 = i, |S_{\ell,t} - t| \leqslant \varepsilon) \cdot \mathbb{P}(|S_{\ell,t} - t| \leqslant \varepsilon) \\ & \quad \geqslant \big(p_{ij}(m,m';t) - \varepsilon \cdot \sigma\big) \cdot \big(1 - \mathbb{P}(|S_{\ell,t} - t| > \varepsilon)\big) \\ & \quad \geqslant p_{ij}(m,m';t) - \varepsilon \cdot \sigma - \Psi_{\ell,t}(\varepsilon). \end{aligned}$$

Pick again $\varepsilon = \ell^{-1/3}$, so as to obtain

$$\liminf_{\ell \to \infty} p_{ij}^{(\mathrm{e},\ell)}(m,m';t) \geqslant p_{ij}(m,m';t).$$

The following theorem summarizes the above findings, thus justifying the use of the Erlangization procedure.

Theorem 1. For any $\ell \in \mathbb{N}$, $t > 0$, and $\varepsilon \in (0,t)$, with σ defined as in (3.2) and $\Psi_{\ell,t}(\varepsilon)$ defined as in (3.10),

$$\Big| p_{ij}^{(\mathrm{e},\ell)}(m,m';t) - p_{ij}(m,m';t) \Big| \leqslant \varepsilon \cdot \sigma + \Psi_{\ell,t}(\varepsilon).$$

In addition, for any $t > 0$,

$$\lim_{\ell \to \infty} p_{ij}^{(\mathrm{e},\ell)}(m,m';t) = p_{ij}(m,m';t).$$

Note that the advantage of Erlangization is that the number of matrix multiplications is low, in comparison with uniformization. More precisely, picking ℓ a power of two, one just needs to square $\Pi_{\ell/t}$ only $\log_2 \ell$ times. The disadvantage is that the computation of the matrix $\Pi_{\ell/t}$ requires the solution of $C+1$ linear systems (the j-th being of dimension $d_j\, D$, $j = 0, 1, \ldots, C$).

3.5 Performance analysis of Erlangization

In this section we examine the performance of the Erlangization approximation of P_t, as given by (3.8). We compare it with the matrix exponential approach given by (3.4) as well as uniformization (3.5). We study the accuracy (i.e., error) and efficiency (i.e., computational time) of the Erlangization approximation. To assess the error we use $P_t^{(\mathrm{m})}$ in (3.4) as benchmark, since the sophisticated implementation `expm`$(\cdot)$ that MATLAB is using is highly accurate and has been tested intensively. In the Erlangization approach we in particular vary ℓ to investigate its influence on the accuracy and efficiency. For the uniformization approach, we use the smallest value of ℓ such that, for the parameters of

the specific example, the Chernoff bound (3.6) is below $\varepsilon = 10^{-4}$. In the sequel we refer to the Erlangization approach by 'E', to the matrix exponential approach by 'M', and to the uniformization approach by 'U'.

In our performance analysis we focus on three QBD processes that are effectively the modulated counterparts of frequently used BD processes. In all three settings the modulating process (also referred to as environmental process) is of dimension 2, irrespectively of the level $m \in \{0, 1, \ldots, C\}$. In other words, we have that $d_m = d = 2$, so that

$$Q^{(m)} = \begin{pmatrix} -q_1 & q_1 \\ q_2 & -q_2 \end{pmatrix}$$

In addition, we let $\lambda_{ij}^{(m)} = 0$ for $i \neq j$, which (informally) means that an increase in level cannot occur at the same time as a phase jump. The three settings are parameterized by a function $f(m, C)$, in the sense that

$$\lambda_i^{(m)} = \lambda_{ii}^{(m)} := f(m, C)\, \lambda_i,$$

for a known positive function $f(m, C)$ and parameter $\lambda_i \geqslant 0$. Similarly, we let $\mu_{ij}^{(m)} = 0$ for $i \neq j$, and define

$$\mu_i^{(m)} = \mu_{ii}^{(m)} := g(m, C)\, \mu_i,$$

for a known positive function $g(m, C)$ and parameter $\mu_i \geqslant 0$. Hence, there are at most six parameters in these models: $q_1, q_2, \lambda_1, \lambda_2, \mu_1$, and μ_2. We proceed by detailing the dynamics underlying the three models.

Experiment 3.1 (Infinite-server queue)**.** Here we consider a system, which can also been seen as a population process, in which individuals arrive according to some arrival process and are served in parallel, in the literature also known as an infinite-server queue [40, 42]. The special feature is that the Poissonian arrival rate as well as the exponential service rate depend on the state of the modulating process, so that the system at hand is a Markov-modulated infinite-server queue [4, 12, 13]. This concretely means that $f(m, C) = 1$ and $g(m, C) = m$ (the latter reflecting that the individuals are served in parallel), with $\Lambda^{(m)} = \mathrm{diag}\{\lambda_1, \lambda_2\}$ and $\mathcal{M}^{(m)} = \mathrm{diag}\{m\,\mu_1, m\,\mu_2\}$. We impose a truncation at level C.

Experiment 3.2 (Linear birth-death process)**.** In this setting we consider the stochastic version of the classical Malthusian growth model, also known as the linear birth-death model [24, 38]: the rate upward as well as the rate downward is proportional to the number of individuals present. This concretely means that $f(m, C) = m$ and $g(m, C) = m$. The rates of moving upward and downward are modulated, which entails that in this case $\Lambda^{(m)} = \mathrm{diag}\{m\,\lambda_1, m\,\lambda_2\}$ and $\mathcal{M}^{(m)} = \mathrm{diag}\{m\,\mu_1, m\,\mu_2\}$. We again impose a truncation at C.

Experiment 3.3 (SIS-type model)**.** The SIR model is a so-called compartmental model used to describe epidemic growth, that keeps track of the number of susceptible individuals, the number of infectious individuals, and the number of recovered individuals; see

e.g. the textbook treatments in [3, 6, 21]. In a related variant, the SIS model, recovered individuals eventually become susceptible again. In this experiment we consider a model of the latter type, which, in the non-modulated context, has the following dynamics. There are C individuals, to be divided into infected and healthy. Let M_t be the number of healthy individuals. When $M_t = m$, an arbitrary healthy person becomes infected with rate $\lambda(C-m)$; as a result the rate from m to $m+1$ is $\lambda\, m(C-m)$. Every infected person becomes healthy again independently of the state of all other individuals; as a result, the rate from m to $m-1$ is $m\,\mu$. If we add modulation, then the λ and μ become dependent on the environmental process. We thus get that in this model $f(m,C) = m(C-m)$ and $g(m,C) = m$, so that the upward rates become $\Lambda^{(m)} = \mathrm{diag}\{m(C-m)\lambda_1, m(C-m)\lambda_2\}$, whereas the downward rates are given by $\mathcal{M}^{(m)} = \mathrm{diag}\{m\,\mu_1, m\,\mu_2\}$.

We start, in Section 3.5.1, with an extensive analysis of Experiment 3.1, the infinite-server queue. In particular we study the impact of the parameters ℓ and C on the accuracy (i.e., error) and efficiency (i.e., computational time) of the Erlangization approximation, and compare these with the other two approaches. In Section 3.5.2 we consider Experiments 3.2 and 3.3.

Importantly, whenever presenting computational times, we report the time it takes to evaluate the entire matrix $P_t^{(\mathrm{e},\ell)}$ ($P_t^{(\mathrm{m})}$ and $P_t^{(\mathrm{u},\ell)}$ likewise), providing us with $p_{ij}^{(\mathrm{e},\ell)}(m,m';t)$ for all $m,m' = 0,\ldots,C$. Furthermore, we use MATLAB's implementation `timeit`$(\cdot)$ to evaluate the computational times (where `timeit`$(\cdot)$ calls the specified function multiple times, measures the time required each time, and finally outputs the median of all these values).

3.5.1 Analysis of Experiment 3.1

We consider Experiment 3.1 with the parameter values $q_1 = 0.015, q_2 = 0.045$, $\lambda_1 = 2, \lambda_2 = 9$ and $\mu_1 = \mu_2 = 0.3$, hence the phase process does not affect the service rate, and let $C = 60$. We compute the transition probability $p_{ij}(m,m';t)$, as defined in (3.3), using the three approaches that we discussed. The results for $i = j = 1$, $m = 10$ and $t = 1$ for varying m', are graphically presented in Figure 3.2. To also get insight into the accuracy for values of m' in which the probability of interest is small, we have in addition included Table 3.1. It shows that Erlangization yields highly accurate results already for $\ell = 128 = 2^7$ (which, remarkably, only requires seven matrix multiplications, besides solving a system of linear equations). The performance slightly degrades when the probability of interest is (extremely) small, say in the range of 10^{-5}–10^{-6}. The last row displays the computational time (in seconds) corresponding to the various values of ℓ, which shows that Erlangization performs well compared with the alternative approaches. Furthermore, we observe that the various computational times for the Erlangization approach are just mildly affected by the value of ℓ. This is explained by the fact that for these values of ℓ, most of the computational time is consumed by the time it takes to solve the system of linear equations that yields $\Pi_{\ell/t}$. In other words, only a small portion of the computational time is due to repeatedly squaring this matrix. For higher values of ℓ we do observe that the computational times grow, but rather slowly.

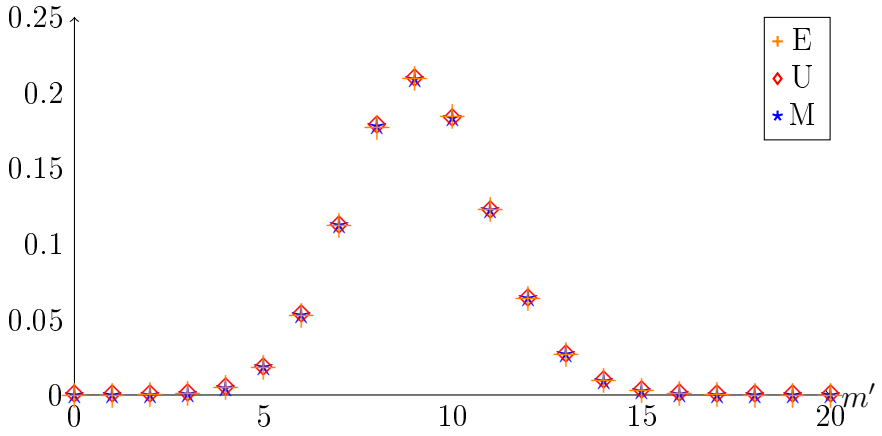

Figure 3.2: *Infinite-server queue*: $p_{ij}(m, m'; t)$ with $i = j = 1$, $m = 10$, $t = 1$ and $C = 60$, computed with the three different methods. Erlangization with $\ell = 128$. Parameter values: $q_1 = 0.015, q_2 = 0.045$, $\lambda_1 = 2, \lambda_2 = 9$ and $\mu_1 = \mu_2 = 0.3$.

Extensive additional experimentation showed that changing the values of the parameters $q_1, q_2, \lambda_1, \lambda_2, \mu_1 = \mu_2$ hardly has any impact on the accuracy of the Erlangization approach. We only saw a slight drop in accuracy when the parameter values were chosen in such a way that (with high probability) m' will be much higher, or much lower, than m. In this respect we refer to Figure 3.3 where $\mu_1 = \mu_2 = 3$, and $m = 50$, with $C = 60$ and $\ell = 128 = 2^7$. Increasing ℓ (as a power of two), for example to $\ell = 1024 = 2^{10}$, such that the exponent increases a few steps, typically returns approximations with a sufficiently high accuracy, while the computational time remains low. This is also why in the sequel we often take this value of ℓ. When we investigate the impact of C on the efficiency, this value of ℓ makes sure that the accuracy is sufficiently high, such that the focus lies completely on the computational time.

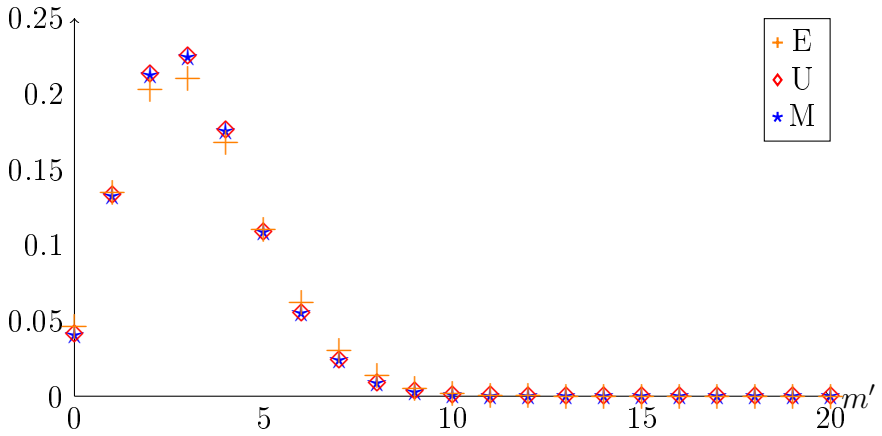

Figure 3.3: *Infinite-server queue*: $p_{ij}(m, m'; t)$ with $i = j = 1$, $m = 50$, $t = 1$ and $C = 60$, computed with the three different methods. Erlangization with $\ell = 128$. Compared to Figure 3.2, we now consider $\mu_1 = \mu_2 = 3$.

m'	E						U	M
	$\ell = 4$	$\ell = 8$	$\ell = 16$	$\ell = 32$	$\ell = 64$	$\ell = 128$		
0	$2.92{\cdot}10^{-6}$	$1.25{\cdot}10^{-6}$	$6.47{\cdot}10^{-7}$	$4.17{\cdot}10^{-7}$	$3.22{\cdot}10^{-7}$	$2.79{\cdot}10^{-7}$	$2.39{\cdot}10^{-7}$	$2.39{\cdot}10^{-7}$
1	$4.41{\cdot}10^{-5}$	$2.35{\cdot}10^{-5}$	$1.46{\cdot}10^{-5}$	$1.07{\cdot}10^{-5}$	$8.91{\cdot}10^{-6}$	$8.07{\cdot}10^{-6}$	$7.25{\cdot}10^{-6}$	$7.26{\cdot}10^{-6}$
2	$3.36{\cdot}10^{-4}$	$2.18{\cdot}10^{-4}$	$1.58{\cdot}10^{-4}$	$1.29{\cdot}10^{-4}$	$1.14{\cdot}10^{-4}$	$1.07{\cdot}10^{-4}$	$1.00{\cdot}10^{-4}$	$1.00{\cdot}10^{-4}$
3	$1.72{\cdot}10^{-3}$	$1.31{\cdot}10^{-3}$	$1.08{\cdot}10^{-3}$	$9.61{\cdot}10^{-4}$	$8.98{\cdot}10^{-4}$	$8.66{\cdot}10^{-4}$	$8.33{\cdot}10^{-4}$	$8.33{\cdot}10^{-4}$
4	$6.56{\cdot}10^{-3}$	$5.74{\cdot}10^{-3}$	$5.24{\cdot}10^{-3}$	$4.96{\cdot}10^{-3}$	$4.81{\cdot}10^{-3}$	$4.73{\cdot}10^{-3}$	$4.65{\cdot}10^{-3}$	$4.65{\cdot}10^{-3}$
5	$1.98{\cdot}10^{-2}$	$1.92{\cdot}10^{-2}$	$1.89{\cdot}10^{-2}$	$1.86{\cdot}10^{-2}$	$1.85{\cdot}10^{-2}$	$1.85{\cdot}10^{-2}$	$1.84{\cdot}10^{-2}$	$1.84{\cdot}10^{-2}$
6	$4.89{\cdot}10^{-2}$	$5.06{\cdot}10^{-2}$	$5.16{\cdot}10^{-2}$	$5.22{\cdot}10^{-2}$	$5.26{\cdot}10^{-2}$	$5.28{\cdot}10^{-2}$	$5.29{\cdot}10^{-2}$	$5.29{\cdot}10^{-2}$
7	$9.91{\cdot}10^{-2}$	$1.05{\cdot}10^{-1}$	$1.09{\cdot}10^{-1}$	$1.10{\cdot}10^{-1}$	$1.11{\cdot}10^{-1}$	$1.12{\cdot}10^{-1}$	$1.12{\cdot}10^{-1}$	$1.12{\cdot}10^{-1}$
8	$1.64{\cdot}10^{-1}$	$1.71{\cdot}10^{-1}$	$1.74{\cdot}10^{-1}$	$1.76{\cdot}10^{-1}$	$1.77{\cdot}10^{-1}$	$1.77{\cdot}10^{-1}$	$1.78{\cdot}10^{-1}$	$1.78{\cdot}10^{-1}$
9	$2.15{\cdot}10^{-1}$	$2.13{\cdot}10^{-1}$	$2.11{\cdot}10^{-1}$	$2.10{\cdot}10^{-1}$	$2.09{\cdot}10^{-1}$	$2.09{\cdot}10^{-1}$	$2.09{\cdot}10^{-1}$	$2.09{\cdot}10^{-1}$
10	$2.07{\cdot}10^{-1}$	$1.94{\cdot}10^{-1}$	$1.89{\cdot}10^{-1}$	$1.86{\cdot}10^{-1}$	$1.85{\cdot}10^{-1}$	$1.84{\cdot}10^{-1}$	$1.83{\cdot}10^{-1}$	$1.83{\cdot}10^{-1}$
11	$1.28{\cdot}10^{-1}$	$1.26{\cdot}10^{-1}$	$1.24{\cdot}10^{-1}$	$1.23{\cdot}10^{-1}$	$1.23{\cdot}10^{-1}$	$1.23{\cdot}10^{-1}$	$1.22{\cdot}10^{-1}$	$1.22{\cdot}10^{-1}$
12	$5.97{\cdot}10^{-2}$	$6.20{\cdot}10^{-2}$	$6.31{\cdot}10^{-2}$	$6.35{\cdot}10^{-2}$	$6.38{\cdot}10^{-2}$	$6.39{\cdot}10^{-2}$	$6.40{\cdot}10^{-2}$	$6.40{\cdot}10^{-2}$
13	$2.34{\cdot}10^{-2}$	$2.50{\cdot}10^{-2}$	$2.59{\cdot}10^{-2}$	$2.64{\cdot}10^{-2}$	$2.67{\cdot}10^{-2}$	$2.68{\cdot}10^{-2}$	$2.69{\cdot}10^{-2}$	$2.69{\cdot}10^{-2}$
14	$8.04{\cdot}10^{-3}$	$8.63{\cdot}10^{-3}$	$8.99{\cdot}10^{-3}$	$9.19{\cdot}10^{-3}$	$9.30{\cdot}10^{-3}$	$9.36{\cdot}10^{-3}$	$9.41{\cdot}10^{-3}$	$9.41{\cdot}10^{-3}$
15	$2.51{\cdot}10^{-3}$	$2.63{\cdot}10^{-3}$	$2.71{\cdot}10^{-3}$	$2.75{\cdot}10^{-3}$	$2.77{\cdot}10^{-3}$	$2.79{\cdot}10^{-3}$	$2.80{\cdot}10^{-3}$	$2.80{\cdot}10^{-3}$
16	$7.29{\cdot}10^{-4}$	$7.29{\cdot}10^{-4}$	$7.27{\cdot}10^{-4}$	$7.26{\cdot}10^{-4}$	$7.25{\cdot}10^{-4}$	$7.24{\cdot}10^{-4}$	$7.23{\cdot}10^{-4}$	$7.23{\cdot}10^{-4}$
17	$2.02{\cdot}10^{-4}$	$1.89{\cdot}10^{-4}$	$1.79{\cdot}10^{-4}$	$1.74{\cdot}10^{-4}$	$1.70{\cdot}10^{-4}$	$1.69{\cdot}10^{-4}$	$1.67{\cdot}10^{-4}$	$1.67{\cdot}10^{-4}$
18	$5.56{\cdot}10^{-5}$	$4.76{\cdot}10^{-5}$	$4.23{\cdot}10^{-5}$	$3.92{\cdot}10^{-5}$	$3.75{\cdot}10^{-5}$	$3.66{\cdot}10^{-5}$	$3.57{\cdot}10^{-5}$	$3.57{\cdot}10^{-5}$
19	$1.66{\cdot}10^{-5}$	$1.27{\cdot}10^{-5}$	$1.03{\cdot}10^{-5}$	$9.03{\cdot}10^{-6}$	$8.36{\cdot}10^{-6}$	$8.01{\cdot}10^{-6}$	$7.65{\cdot}10^{-6}$	$7.65{\cdot}10^{-6}$
20	$6.02{\cdot}10^{-6}$	$4.02{\cdot}10^{-6}$	$2.96{\cdot}10^{-6}$	$2.43{\cdot}10^{-6}$	$2.15{\cdot}10^{-6}$	$2.02{\cdot}10^{-6}$	$1.88{\cdot}10^{-6}$	$1.88{\cdot}10^{-6}$
CPU times	$1.00{\cdot}10^{-2}$	$8.00{\cdot}10^{-3}$	$9.35{\cdot}10^{-3}$	$1.08{\cdot}10^{-2}$	$1.19{\cdot}10^{-2}$	$1.30{\cdot}10^{-2}$	$7.19{\cdot}10^{-2}$	$2.51{\cdot}10^{-2}$

Table 3.1: *Infinite-server queue*: $p_{ij}(m, m'; t)$ with $i = j = 1$, $m = 10$, $t = 1$ and $C = 60$, and CPU times corresponding to the approximation of P_t. Parameter values: $q_1 = 0.015, q_2 = 0.045$, $\lambda_1 = 2, \lambda_2 = 9$ and $\mu_1 = \mu_2 = 0.3$.

Evidently, computational times increase in C; we proceed by investigating this relation for the three methods. We do so by increasing C from 25 to 500 in steps of 25 and 50. The results are shown in Figure 3.4. We see that the computational times for the matrix exponential method and Erlangization are essentially of the same order. For small values of C, the Erlangization method is faster, whereas for higher values of C the matrix exponential method is faster. The computational times for the uniformization method, however, are significantly longer. This is in line with what we expected, since uniformization typically needs a large number of matrix multiplications.

To systematically assess the impact of C on the computational time, which we denote by T, we fit the curve $T = \alpha C^{\beta}$. This we do by applying least squares to $\log T = \log \alpha + \beta \log C$. We thus find that the time of Erlangization is close to quadratic in C ($\beta = 2.12$), the CPU time of the matrix exponential method is subquadratic in C ($\beta = 1.59$), whereas the CPU time of uniformization is superquadratic in C ($\beta = 2.40$); see also Figure 3.4.

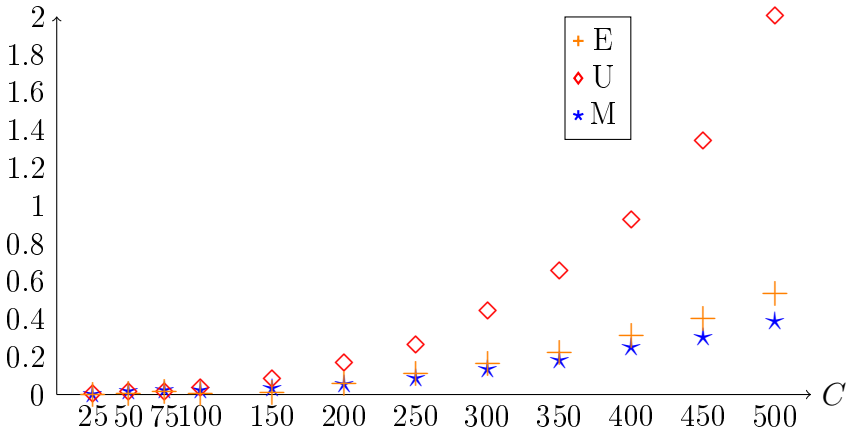

Figure 3.4: *Infinite-server queue*: CPU times (in seconds) corresponding to the approximation of P_t with $t = 1$, for the three different methods. Erlangization with $\ell = 1024$. Parameter values: $q_1 = 0.015, q_2 = 0.045$, $\lambda_1 = 2, \lambda_2 = 9$ and $\mu_1 = \mu_2 = 0.3$.

3.5.2 Other experiments

To explore if other settings yield similar results, we investigate the two other experiments as well. We consider Experiment 3.2 with parameter values $q_1 = 0.3, q_2 = 0.9$, $\lambda_1 = \lambda_2 = 0.19$, $\mu_1 = 0.16$, $\mu_2 = 0.08$ (i.e., the phase process does not affect the birth rate) and $C = 300$, and we consider Experiment 3.3 with parameter values $q_1 = 0.1, q_2 = 0.4$, $\lambda_1 = 0.0035$, $\lambda_2 = 0.01$, $\mu_1 = \mu_2 = 0.3$ (i.e., the phase process does not affect the recovery rate) and $C = 100$. We briefly present the results, focusing on the differences with the results of Experiment 3.1.

We first revisit the accuracy of the Erlangization approximation of $p_{ij}(m, m'; t)$ and the influence of ℓ on the accuracy. We conclude that the accuracy achieved in Experiments 3.2 and 3.3 strongly resembles the accuracy of Experiment 3.1, across a broad range of values of ℓ. Already for small ℓ the approximations are highly accurate, and the accuracy

improves as ℓ increases. Figures 3.5 and 3.6 show the counterparts of Figure 3.2, with $\ell = 128$, $i = j = 1$, $m = 10$ and $t = 1$ for varying m' as before, and illustrate the similarity in accuracy between the three experiments.

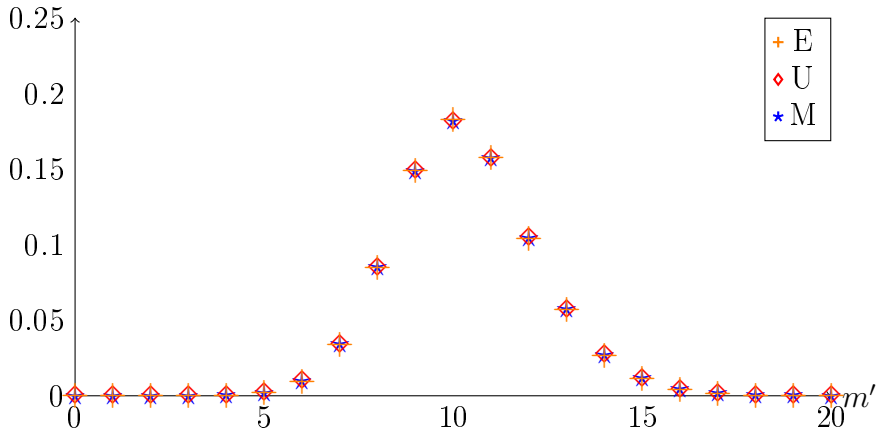

Figure 3.5: *Linear birth-death process*: $p_{ij}(m, m'; t)$ with $i = j = 1$, $m = 10$, $t = 1$ and $C = 300$, computed with the three different methods. Erlangization with $\ell = 128$. Parameter values: $q_1 = 0.3, q_2 = 0.9$, $\lambda_1 = \lambda_2 = 0.19$, $\mu_1 = 0.16$ and $\mu_2 = 0.08$.

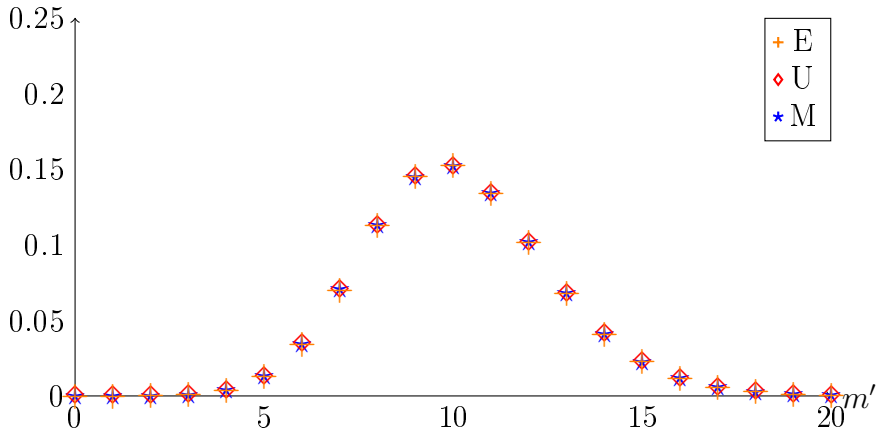

Figure 3.6: *SIS-type model*: $p_{ij}(m, m'; t)$ with $i = j = 1$, $m = 10$, $t = 1$ and $C = 100$, computed with the three different methods. Erlangization with $\ell = 128$. Parameter values: $q_1 = 0.1, q_2 = 0.4$, $\lambda_1 = 0.0035$, $\lambda_2 = 0.01$ and $\mu_1 = \mu_2 = 0.3$.

We now examine the impact of C on the computational time, with $t = 1$. Figure 3.7 shows for each specific approximation the computational times corresponding to the three experiments. The main conclusion from Figure 3.7 is that the observations for Experiment 3.2 to a large extent agree with those for Experiment 3.1, but that Experiment 3.3 behaves rather differently. For the matrix exponential approach and Erlangization approach, the computational times corresponding to the three experiments roughly coincide. However, for uniformization the computation times for Experiment 3.3 strongly deviate from the computational times of the other experiments.

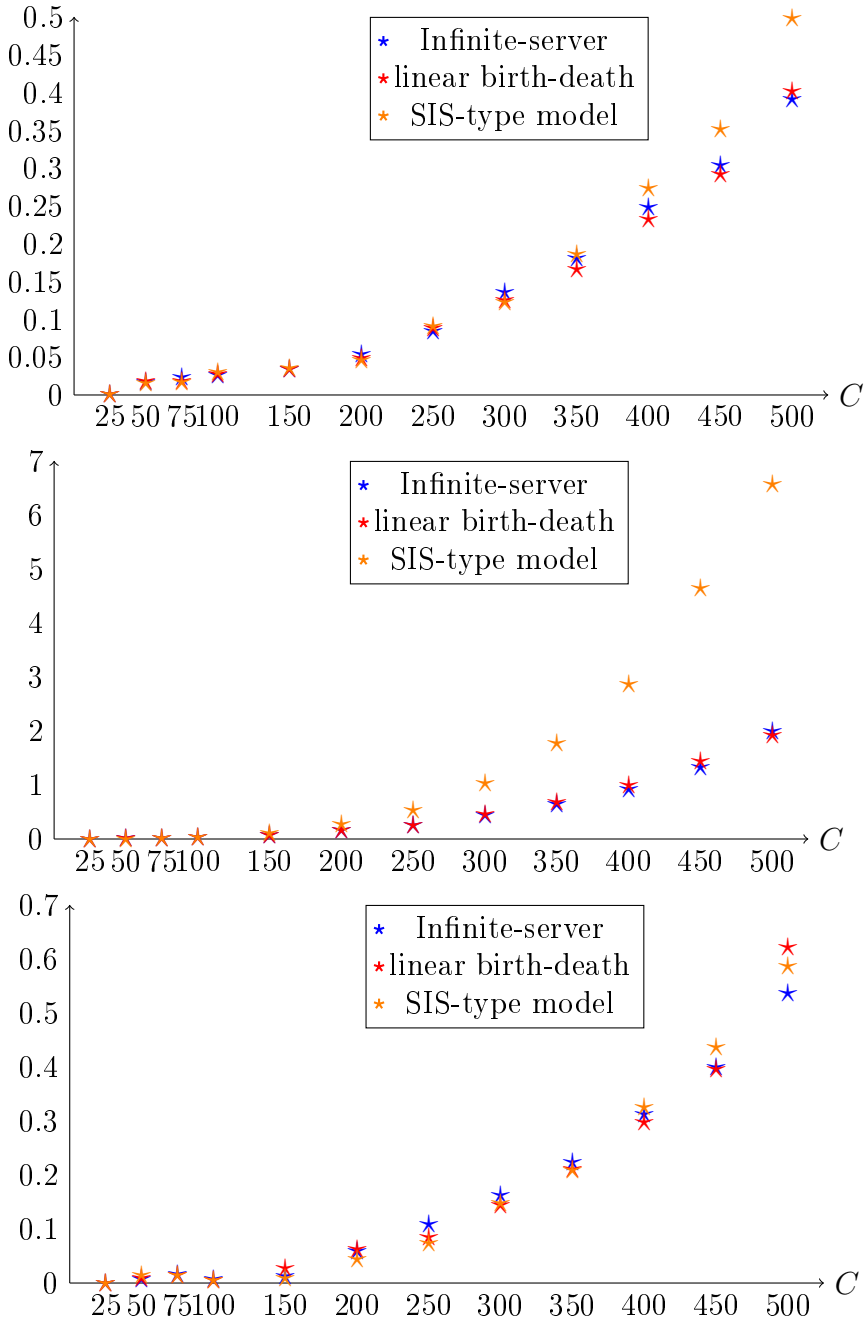

Figure 3.7: CPU times (in seconds) measured for the three experiments and the three different methods; from the top to bottom panel, Matrix exponential method, Uniformization method and Erlangization method with $\ell = 1024$. Parameter values infinite-server queue: $q_1 = 0.015, q_2 = 0.045$, $\lambda_1 = 2, \lambda_2 = 9$, $\mu_1 = \mu_2 = 0.3$ and $C = 60$. Parameter values linear birth-death process: $q_1 = 0.3, q_2 = 0.9$, $\lambda_1 = \lambda_2 = 0.19$, $\mu_1 = 0.16$, $\mu_2 = 0.08$ and $C = 300$. Parameter values SIS-type model: $q_1 = 0.1, q_2 = 0.4$, $\lambda_1 = 0.0035$, $\lambda_2 = 0.01$, $\mu_1 = \mu_2 = 0.3$ and $C = 100$.

When fitting the curve $T = \alpha C^{\beta}$, the difference described above becomes clearly visible. Table 3.2 shows the computed values of β for the different experiments and the different approximation approaches. Erlangization and the matrix exponential method lead to roughly the same β in all three experiments. This reflects that the computational time is essentially determined by the value of C, and is not affected by the choice of $f(m, C)$ and $g(m, C)$, and hence not by the magnitude of the entries. For uniformization, however, the number of terms needed strongly depends on the magnitude of the entries of the Q-matrix, leading to a stronger growth of the computational time for the SIS-type model.

Remark 1. The fact that uniformization is slow for the SIS-type model can be understood as follows. The number of terms needed in (3.5), which in turn determines the number of matrix multiplications to be performed, is σt increased by some margin that makes sure that $\mathbb{P}(\text{Pois}(\sigma t) \leqslant \ell + 1)$ is sufficiently small. Recall that σ is the (absolute value of) the largest diagonal entry of Q. For the infinite-server model and the linear birth-death model, this largest entry is of the order C. For the SIS-type model, however, recalling that $f(m, C) = m(C - m)$, the largest entry is of the order C^2. As a consequence, the number of terms in (3.5) is relative large, leading to a relatively long computational time.

Experiment	E	U	M
Infinite-server	2.1199	2.4038	1.5916
linear birth-death	2.0833	2.5318	1.5911
SIS-type model	2.0593	3.1206	1.6576

Table 3.2: Table with β values for the different experiments and different approaches. Erlangization with $\ell = 1024$. Parameter values as in Figure 3.7.

3.6 Model selection

We started our chapter with a motivating example: can we statistically distinguish whether data stems from a QBD or from its non-modulated counterpart? We argued that to answer this question, we need machinery to evaluate the likelihood corresponding to a given time series. Now that we have at our disposal techniques to evaluate probabilities of the type (3.3), we return to our model selection problem of distinguishing between QBD processes and conventional (non-modulated, that is) BD processes. In this section we do so, using both simulated data and real-life data.

We wish to distinguish between the following four scenarios:

1. No modulation on neither the birth rate λ nor the death rate μ, i.e., $\theta = (\lambda, \mu)$
2. Modulation on the birth rate λ only ($\mu_1 = \mu_2$), i.e., $\theta = (q_1, q_2, \lambda_1, \lambda_2, \mu)$
3. Modulation on the death rate μ only ($\lambda_1 = \lambda_2$), i.e., $\theta = (q_1, q_2, \lambda, \mu_1, \mu_2)$

4. Modulation on both the birth rate λ and the death rate μ, i.e., $\theta = (q_1, q_2, \lambda_1, \lambda_2, \mu_1, \mu_2)$

We start by considering the setting of Experiment 3.1 with simulated data, and then use the model of Experiment 3.2 to analyze the whooping crane data featured in the introduction. We investigate which of these scenarios provides the best fit for the data, using the commonly used Akaike information criterion. This criterion includes a penalty that equals twice the number of estimated parameters (i.e., two times 2, 5, 5, and 6 in the above four scenarios), thus preventing overfitting from happening.

In all experiments below there is a time interval $\Delta > 0$ so that the observations correspond to measurements performed at times $0, \Delta, 2\Delta, \ldots, n\Delta$ for some $n \in \mathbb{N}$. We call these observations $m_0, \ldots, m_n$. With θ the vector of parameters, the likelihood is

$$\mathcal{L}(\theta \,|\, m_0, \ldots, m_n) = \mathbb{P}_\theta(M_0 = m_0, \ldots, M_{n\Delta} = m_n). \tag{3.12}$$

Regarding scenarios 2, 3, and 4, note that the modulating process is not observed. However, with $\boldsymbol{x} = (x_0, \ldots, x_n) \in \{1,2\}^{n+1}$, we can rewrite (3.12) as

$$\begin{aligned} &\sum_{\boldsymbol{x} \in \{1,2\}^{n+1}} \mathbb{P}_\theta(M_0 = m_0, X_0 = x_0, \ldots, M_{n\Delta} = m_n, X_{n\Delta} = x_n) \\ &\qquad = \sum_{\boldsymbol{x} \in \{1,2\}^{n+1}} \prod_{i=1}^{n} p_{x_{i-1},x_i}(m_{i-1}, m_i; \Delta), \end{aligned} \tag{3.13}$$

where it is noted that the probabilities in the last expression are of the type (3.3), and can be evaluated with the techniques discussed in this chapter. Importantly, there is no need to enumerate all paths $\boldsymbol{x} \in \{1,2\}^{n+1}$. Instead we can evaluate (3.13) efficiently by, abbreviating $p_{x_{i-1},x_i}(m[i]) \equiv p_{x_{i-1},x_i}(m_{i-1}, m_i; \Delta)$, evaluating the matrix product

$$\boldsymbol{\alpha} \begin{pmatrix} p_{11}(m[1]) & p_{12}(m[1]) \\ p_{21}(m[1]) & p_{22}(m[1]) \end{pmatrix} \begin{pmatrix} p_{11}(m[2]) & p_{12}(m[2]) \\ p_{21}(m[2]) & p_{22}(m[2]) \end{pmatrix} \cdots \begin{pmatrix} p_{11}(m[n]) & p_{12}(m[n]) \\ p_{21}(m[n]) & p_{22}(m[n]) \end{pmatrix} \mathbf{1}, \tag{3.14}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ is the distribution of X_0 and $\mathbf{1}$ is an all-ones vector. Note that the matrices in (3.14) are given as blocks in $P_t^{(\mathrm{e},\ell)}$. Maximization of the likelihood gives us the maximum likelihood estimate $\hat{\theta}$ for θ. As we will discuss below, this likelihood can be used in model selection problems. In the experiments below, all calculations involving probabilities of the type $p_{x_{i-1},x_i}(m[i])$ have been performed by the Erlangization approach.

3.6.1 Simulated data

We consider the setting of Experiment 3.1. We simulate data ($n = 2000$) with parameter values $q_1 = 0.015, q_2 = 0.045, \lambda_1 = 0.2, \lambda_2 = 0.9, \mu = 0.03$, $\Delta = 1$ and $C = 50$. This

means that the true model for this data is an infinite-server queue with modulation on λ only. Based on this simulated data, we perform the model selection based on the Akaike information criterion, i.e., using $\text{AIC} = 2N - 2\log L(\hat{\theta})$, with N the dimension of the parameter vector θ.

parameter	scenario			
	1.	2.	3.	4.
$\hat{q}_1$	n/a	0.0120	0.0953	0.0122
$\hat{q}_2$	n/a	0.0456	0.0357	0.0462
$\hat{\lambda}_1$ (or λ)	0.3373	0.2093	0.3374	0.2097
$\hat{\lambda}_2$	n/a	0.8904	n/a	0.8790
$\hat{\mu}_1$ (or μ)	0.0302	0.0312	0.0175	0.0314
$\hat{\mu}_2$	n/a	n/a	0.0361	0.0299
$\log L(\hat{\theta})$	−2370.1	−2306.5	−2368.2	−2306.4
AIC	4744.1	4622.9	4746.4	4624.8

Table 3.3: *Experiment 3.1, simulated data*: parameter estimates, loglikelihood value and AIC for the four different scenarios ($n = 2000$), with $\ell = 1024$ and $C = 50$. True parameter values: $q_1 = 0.015, q_2 = 0.045, \lambda_1 = 0.2, \lambda_2 = 0.9, \mu = 0.03$ with $\Delta = 1$.

From Table 3.3 we observe that the AIC value is smallest for scenario 2, which agrees with the ground truth of the simulated data (i.e., it succeeds in finding the scenario with modulation on the parameter λ only). Interestingly, the number of observations has impact on the conclusions drawn. To illustrate this, see Table 3.4 showing the results using the first 101 data points of the dataset only (i.e., $n = 100$ instead of $n = 2000$). The AIC value is now minimized by scenario 1, the scenario without modulation, indicating that the dataset is too short to detect the modulation.

parameter	scenario			
	1.	2.	3.	4.
$\hat{q}_1$	n/a	0.5265	0.5484	$2.4 \cdot 10^{-7}$
$\hat{q}_2$	n/a	0.5548	0.5715	0.7045
$\hat{\lambda}_1$ (or λ)	0.2351	0.2343	0.2351	0.2351
$\hat{\lambda}_2$	n/a	0.2360	n/a	0.5651
$\hat{\mu}_1$ (or μ)	0.0281	0.0281	0.0280	0.0281
$\hat{\mu}_2$	n/a	n/a	0.0282	0.0769
$\log L(\hat{\theta})$	−104.10	−104.10	−104.10	−104.10
AIC	212.20	218.20	218.20	220.20

Table 3.4: *Experiment 3.1, simulated data*: parameter estimates, loglikelihood value and AIC for the four different scenarios ($n = 100$), with $\ell = 1024$ and $C = 50$. True parameter values: $q_1 = 0.015, q_2 = 0.045, \lambda_1 = 0.2, \lambda_2 = 0.9, \mu = 0.03$ with $\Delta = 1$.

3.6.2 Whooping crane population

We proceed by considering the linear birth-death setting of Experiment 3.2 in relation to the four scenarios mentioned above. We use the whooping crane data [24, 66], as displayed in Figure 3.1, of annual counts of the female population of the whooping crane $n = 69$. From Figure 3.1 we could suspect that a model with modulation could lead to a better fit than a model without modulation. We (conservatively) set $C = 200$. The outcomes of the model selection procedure are shown in Table 3.5. As it turns out, the AIC value is smallest for scenario 1, i.e., the setting corresponding with no modulation. One should bear in mind, though, that the number of observations in this dataset is low, making the detection of modulation (involving 5 or 6 parameters) difficult. Additional literature on parameter estimation for linear birth-death models can be found in e.g. [18, 19, 20, 24, 70].

parameter	scenario			
	1.	2.	3.	4.
$\hat{q}_1$	n/a	0.9338	0.7948	0.9504
$\hat{q}_2$	n/a	0.2034	0.5084	0.1571
$\hat{\lambda}_1$	0.1928	0.1588	0.1789	0.1206
$\hat{\lambda}_2$	n/a	0.1966	n/a	0.1889
$\hat{\mu}_1$	0.1492	0.1462	0.0969	$3.7 \cdot 10^{-7}$
$\hat{\mu}_2$	n/a	n/a	0.1603	0.1574
$\log L(\hat{\theta})$	−179.66	−179.65	−179.57	−179.40
AIC	363.32	369.31	369.14	370.80

Table 3.5: *Whooping crane data*: parameter estimates, loglikelihood value and AIC for the four different scenarios ($n = 69$), with $\ell = 1024$ and $C = 200$.

3.7 Concluding remarks

We have examined various approaches to compute the tine-dependent distribution of QBD processes, with emphasis on the Erlangization approach. This approach has provable asymptotic correctness properties, and is, in terms of computational time, typically relatively fast. The latter property pays off in particular in settings where many time-dependent probabilities have to be evaluated. In this context, one could think of instances in which a function of the time-dependent probabilities is to be optimized over a set of model parameters, e.g. when performing maximum likelihood estimation.

Our study was motivated by model selection problems, in which one wishes to distinguish between models with and without modulation, i.e., between QBD processes and their BD counterparts. Through a series of experiments, with simulated as well as real-life data, we have shown how the techniques for computing time-dependent distributions can play a role in this context.

Our Erlangization approach gives rise to various directions for further research. For

the class of QBD processes, the method's first step (solving the system of linear equations that yield the probabilities at exponential epochs) can exploit the convenient underlying structure, thus allowing an efficient numerical algorithm. We anticipate, however, that Erlangization has the potential to be applied more widely. One could think of multi-type population models, where various types of individuals are considered, which can in turn interact with each other. Another interesting extension concerns the multivariate model in which a population of individuals lives on a network and can move between its nodes. In this respect we refer to Chapter 5, approximating time-dependent probabilities in such a network, relying on saddlepoint approximations. The crucial simplification made in Chapter 5 is that a discrete-time model is considered, as opposed to the continuous-time model featuring in the present chapter. It would therefore be interesting to explore whether an Erlangization-based approach could be developed for the continuous-time setting of such a network population process.

4. Population model for mRNA transcription

A birth-death process of which the births follow a hypoexponential distribution with L phases and are controlled by an on/off mechanism, is a population process which we call the on/off-seq-L process. It is a suitable model for the dynamics of a population of mRNA molecules in single living cells. Motivated by this biological application, a method is presented to compute maximum likelihood estimates of the model parameters, based on observations of the population size at discrete time points. It is shown that the on/off-seq-L process can be seen as a quasi birth-death process, and the Erlangization technique can be used to approximate the likelihood function. To investigate the performance of the resulting estimation method, an extensive simulation-based numerical study is carried out. Numerical complications related to the likelihood maximization are analyzed and solutions are presented. The estimation method is applied to real mRNA data, and a model selection procedure is performed on the number of phases and the on/off mechanism in the on/off-seq-L process.

4.1 Introduction

Birth-death (BD) processes are continuous-time Markov processes with two types of transitions; *births* which increase the state by one, and *deaths* which decrease the state by one. BD processes are suitable to model the dynamics of the number of individuals in a population, and are widely used in a broad range of areas such as biology, ecology and operations research. The research in this chapter is motivated by a specific biological application: the number of mRNA molecules in a single living cell. The evolution of a population of mRNA molecules can be modelled by a BD process, since the population can increase (production) or decrease (degradation) by one molecule at a time. However, it is known that the production of mRNA molecules is a sequential process consisting of multiple phases [44, 64], and that the production is regulated by an on/off mechanism [53], which we will refer to as the on/off switch. To model the population of mRNA molecules in a realistic way, we therefore extend the basic BD process by including these two features to the model. This results in what we call the on/off-seq-L process, which is also considered in [30]. The on/off switch in the on/off-seq-L process is a mechanism that decides if the next birth of an individual can be set in motion or not. Births can be initiated only while the switch is turned on. If the switch turns off, it needs to be switched back on before

a birth can be initiated. Once a birth is initiated, it takes L sequential exponentially distributed phases before a new individual is born and the population increases by one.

We are interested in performing statistical inference for the on/off-seq-L process, and its application to a real data set of mRNA counts in cells. Motivated by the structure of this real data set, we consider an inference problem based on a data set that consists of multiple time series, which is a broadly applicable data setting. The goal is to estimate the model parameters based on observations of the population size at discrete time points, and to perform model selection on the on/off switch and on the number of phases L in the birth process. This kind of inference problem has been studied before in the context of mRNA transcription. We mention [37], where maximum likelihood estimates are computed and a model selection procedure is performed for a stochastic model with a sequential birth process. However, in contrast to the on/off-seq-L process, an on/off mechanism is not included in that model. In [30, 51], maximum likelihood estimation and a model selection procedure are performed for the on/off-seq-L process. However, in these studies the likelihood function is computed from observations of the transcription intervals, that is, the time between two consecutive mRNA births. These intervals are not observed precisely and censoring is needed to compute the likelihood function. In this chapter, we make use of the fact that the on/off-seq-L model is actually a special case of a quasi birth-death (QBD) process. This means that the Erlangization technique introduced in Chapter 3 can be applied to evaluate the likelihood function from the population size, instead of the transcription intervals.

The remainder of this chapter is organized as follows. In Section 4.2, we mathematically define the on/off-seq-L process and introduce the corresponding likelihood function and estimation problem. Section 4.3 shows that the on/off-seq-L process belongs to the class of QBD processes, and therefore the Erlangization method introduced in Chapter 3 can be used to approximate the likelihood. By an extensive numerical study in Section 4.4, we investigate the accuracy of the resulting estimation method for the on/off-seq-L process. In addition, we explore numerical complications related to the likelihood maximization. Section 4.5 describes in detail the biological process of mRNA transcription, which is the motivating application of this chapter. A model selection procedure is performed for different on/off-seq-L processes, based on data of mRNA counts in single cells. The chapter is concluded by a discussion in Section 4.6.

4.2 Mathematical model and estimation problem

In this section we formally introduce the class of on/off-seq-L processes together with the necessary notation. We then define the estimation problem and the corresponding likelihood function.

4.2.1 The on/off-seq-L process

The on/off-seq-L process can be viewed as a BD process with two specific features in the birth process. First, the births follow a hypoexponential distribution—that is a sum of exponentially distributed phases—instead of the often used exponential distribution. Second, the births are controlled by a so-called on/off switch, which means that births can be initiated only while the switch is turned on. Because of this specific structure, the on/off-seq-L process is modelled as a two-dimensional Markov process, consisting of the population process together with an underlying background process. We start with the mathematical definition of this background process, which can be viewed as a process that keeps track of the status of the birth process. We then define the population process and complete the definition with the two-dimensional Markov process and its transition rates.

Let $\{X_t\}_{t\geq 0}$ be a continuous-time Markov chain modeling both the on-off switch of the process and the exponential phases of the birth process. Its state space is given by $E = \{0, 1, \ldots, L\}$. The state $X_t = 0$ corresponds to the state where the on/off switch is turned off, and will be referred to as the off-state. Importantly, births cannot be initiated in this state. The switch needs to switch back on first, leading to the state $X_t = 1$, which we refer to as the on-state. Births can only be initiated from this state. Once a birth is initiated, the process runs through states $1, \ldots, L$ and back to state 1, corresponding to the sequential, exponential phases of the birth process. A schematic representation is given in Figure 4.1 for the model with $L = 3$. When the L exponential phases are completed, a new individual is born and the population increases by one. During this birth process, the switch remains on.

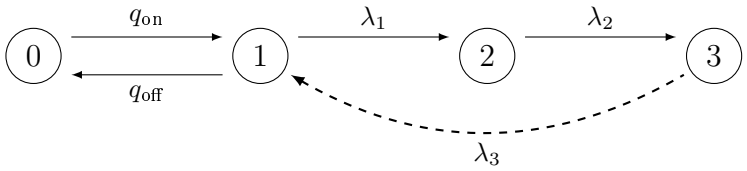

Figure 4.1: Schematic representation of the $\{X_t\}$ process in the on/off-seq-3 model. The dotted line indicates the transition that results in a birth of a new individual. Parameters $q_{\text{off}}, q_{\text{on}}, \lambda_1, \lambda_2$ and λ_3 denote the transition rates.

Let $\{M_t\}_{t\geq 0}$ be the population process, with M_t equal to the total number of individuals in the system at time t. The birth process that increases the population size is described above. The population size decreases according to a general death process, where the lifetimes of the individuals are assumed to follow an exponential distribution, independently of each other, and independently of $\{X_t\}$. The entire model is described by the two-dimensional, time-homogeneous Markov process $\{X_t, M_t\}_{t\geq 0}$. Combining the definitions of $\{X_t\}$ and $\{M_t\}$, we can define the transition rates of this joint process.

First, we have the two rates associated with the on-off mechanism. These rates correspond to jumps of X_t between states 0 and 1 while the state of M_t remains unchanged. When $M_t = m$, we have, for all $m \geq 0$, the transition rate q_{on} for the transition from

$(0, m)$ to $(1, m)$ and the rate q_{off} for the transition from $(1, m)$ to $(0, m)$. Note that q_{on} and q_{off} do not depend on m. Secondly, we have the rates associated with the sequential birth phases, where the state of M_t remains unchanged until the completion of the final phase. For all $L \geq 2$ we have rates λ_i for the transitions from (i, m) to $(i+1, m)$, $i \in 1, \ldots, L-1$, and for all $L \geq 1$ we have rate λ_L for the transition (L, m) to $(1, m+1)$. Note that after completion of the final phase, the process $\{X_t\}$ returns to state 1 from which the system can either be turned off, or a new birth can be initiated. Last, we have the rates associated with the deaths. The lifetimes of the individuals follow an exponential distribution with parameter μ, independently of each other. This means that the total death rate is proportional to the total number of individuals in the population. Furthermore, the lifetimes are not affected by the state of $\{X_t\}$. Hence for all $i \in 1, \ldots, L$ and $m > 0$, we have rate $m\mu$ for the transition (i, m) to $(i, m-1)$.

4.2.2 Likelihood evaluation

We combine all model parameters of the on/off-seq-L process in the parameter vector $\theta = (q_{\text{on}}, q_{\text{off}}, \lambda_1, \ldots, \lambda_L, \mu)^\top$. As mentioned above, the goal is to estimate θ based on observations of the population size at discrete time points, and to perform model selection on the on/off switch and on the number of phases L in the birth process. To find maximum likelihood estimates, we need a reliable method to evaluate the likelihood function of the data with respect to θ.

The available data set consists of multiple times series corresponding to N independent experiments. Let $\Delta > 0$ be the time between two consecutive observations, and let $n+1$ be the number of observations in a single experiment corresponding to observation times $0, \Delta, 2\Delta, \ldots, n\Delta$. We assume that in each experiment the process $\{M_t\}$ is observed at these observation times, resulting in observations $m_0^{(k)}, \ldots, m_n^{(k)}$ for experiments $k = 1, \ldots N$. We introduce the corresponding data vectors $m_{0,n}^k = (m_0^{(k)}, \ldots, m_n^{(k)})^\top$, $k = 1, \ldots N$. The loglikelihood function based on the N independent experiments is then equal to

$$\log \mathcal{L}(\theta | m_{0,n}^{(1)}, \ldots, m_{0,n}^{(N)}) = \sum_{k=1}^{N} \log \mathcal{L}(\theta | m_{0,n}^{(k)}). \tag{4.1}$$

We can rewrite the likelihood function, $\mathcal{L}(\theta | m_{0,n}^{(k)})$, for a single data vector $m_{0,n}^k$, by conditioning on the states of the background process $\{X_t\}$ at the observation times. To this end, we define the transition probabilities

$$p_{xx'}(m, m'; t) := \mathbb{P}(M_t = m', X_t = x' | M_0 = m, X_0 = x).$$

Then

$$\mathcal{L}(\theta|m_{0,n}^{(k)}) = \sum_{x_0,\ldots,x_n \in E} \prod_{i=1}^{n} p_{x_{i-1}x_i}(m_{i-1}^{(k)}, m_i^{(k)}; \Delta). \tag{4.2}$$

In the next section we show that the on/off-seq-L process can be seen as a QBD process. This means that the Erlangization technique introduced in Chapter 3 can be applied to approximate the transition probabilities in (4.2), and hence the likelihood function (4.1). A requirement to apply the Erlangization technique is that the population size M_t is bounded from above by a constant $C > 0$. By the nature of the BD process, the state of M_t can only increase by one at a time. This means that for each data vector $m_{0,n}^{(k)}$, we can choose a constant $C \in \mathbb{N}$ large enough such that for all $x, x' \in E$, $m_i^{(k)} < m'$ and $i = 1, \ldots, n$, the transition probability $p_{xx'}(m_i^{(k)}, m'; \Delta)$ is negligible for $m' > C$. Hence, we can bound the population size by this constant C.

4.3 Quasi birth-death framework

In this section we show that the on/off-seq-L process belongs to the class of QBD processes, using the framework as described in Chapter 3. As argued in the previous section, we can assume that the population process $\{M_t\}$ attains values in $\{0, 1, \ldots, C\}$ for some $C > 0$.

Let, as in Chapter 3, $Q^{(m)}$, $m = 0, \ldots, C$, be the transition rate matrix on state space $E = \{0, 1, \ldots, L\}$ corresponding to all jumps of X_t that leave the state of $M_t = m$ unchanged. Note that, in the setting of this chapter, $Q^{(m)}$ is actually independent of m. For example, for $L = 3$ and all $m \in \{0, 1, \ldots, C\}$, we have

$$Q^{(m)} = \begin{pmatrix} -q_{\text{on}} & q_{\text{on}} & 0 & 0 \\ q_{\text{off}} & -q_{\text{off}} - \lambda_1 & \lambda_1 & 0 \\ 0 & 0 & -\lambda_2 & \lambda_2 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

For $i \neq j$, each element $[Q^{(m)}]_{ij}$ corresponds to the jump from $X_t = i$ to $X_t = j$ while $M_t = m$, and the diagonal elements $[Q^{(m)}]_{ii}$ are such that the row sums are zero.

Next, we introduce the matrix $\Lambda^{(m)}$ on E, of which the elements correspond to the jumps that increase M_t by one, while X_t jumps from state i to j. Note that for the on/off-seq-L process, all $\lambda_{ij}^{(m)}$ are zero except for the one corresponding to the completion of the final phase of the birth process (if $m \leq C - 1$). Hence for $L = 3$, and $m \leq C - 1$,

we have

$$\Lambda^{(m)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \lambda_3 & 0 & 0 \end{pmatrix}.$$

At last, we introduce the matrix $\mathcal{M}^{(m)}$ on E, of which the elements correspond to the jumps that decrease M_t by one, while X_t jumps from state i to j. Deaths leave the state of the background process unchanged, hence all $\mu_{ij}^{(m)}$ are zero for $i \neq j$. We have

$$\mathcal{M}^{(m)} = \begin{pmatrix} m\mu & 0 & 0 & 0 \\ 0 & m\mu & 0 & 0 \\ 0 & 0 & m\mu & 0 \\ 0 & 0 & 0 & m\mu \end{pmatrix}.$$

We observe that we can write down the transition rate matrix of the joint process $\{X_t, M_t\}$ in terms of the matrices $Q^{(m)}$, $\Lambda^{(m)}$ and $\mathcal{M}^{(m)}$ in the same way as in Chapter 3. The total number of states of $\{X_t, M_t\}$ is $D = (L+1)(C+1)$, and the $D \times D$ transition matrix is equal to

$$Q := \begin{pmatrix} \bar{Q}^{(0)} & \Lambda^{(0)} & 0 & \cdots & 0 & 0 \\ \mathcal{M}^{(1)} & \bar{Q}^{(1)} & \Lambda^{(1)} & \cdots & 0 & 0 \\ 0 & \mathcal{M}^{(2)} & \bar{Q}^{(2)} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \cdots & \bar{Q}^{(C-1)} & \Lambda^{(C-1)} \\ 0 & 0 & 0 & \cdots & \mathcal{M}^{(C)} & \bar{Q}^{(C)} \end{pmatrix},$$

where $\bar{Q}^{(m)}$ is defined as $Q^{(m)}$ with the diagonal entries adapted such that the row sums of Q are zero. This means that, in contrast to $Q^{(m)}$, the diagonal entries of $\bar{Q}^{(m)}$ depend on m.

We conclude that the on/off-seq-L process can be seen as a special case of a QBD process. This means that we can use the results in Chapter 3 to approximate our likelihood function in a reliable and accurate way. Using the Erlangization technique we can approximate the likelihood $\mathcal{L}(\theta|m_{0,n}^{(k)})$ corresponding to a single data vector $m_{0,n}^{(k)}$ as given in (4.2), which in turn can be used to approximate the likelihood function (4.1) corresponding to N independent experiments. The maximum likelihood estimate $\hat{\theta}$ of θ can be obtained by numerical optimization of the likelihood over the domain $\mathcal{D}$ of θ.

4.4 Numerical study

In this section we investigate the accuracy of the estimation method for the on/off-seq-L process as described above, by means of a simulation-based numerical study. In addition, we identify numerical complications related to the likelihood maximization that we need to take into account, and investigate how to solve them.

Each model setting considered in this section corresponds to a fixed number of phases L and to a fixed parameter vector $\theta = (q_{\text{on}}, q_{\text{off}}, \lambda_1, \ldots, \lambda_L, \mu)^\top$. In our simulation studies, the model setting and the size of the data were chosen first, by fixing L and θ, and fixing n and N. Next, the data vectors $m_{0,n}^k$, for $k = 1, \ldots, N$, were simulated B times, for $B > 0$ large and the estimation method was applied to each of the B groups of data vectors. Here the parameter ℓ in the Erlangization approximation was fixed at $\ell = 2048$ and the domain $\mathcal{D}$ was chosen as $[0, b]^{L+3}$ for a fixed upper bound $b > 0$. This resulted in B estimates for the parameter vector θ, which we denote by $\hat{\theta}_i$, $i = 1, \ldots, B$. By analyzing these parameter estimates, we obtained insight in the performance of the estimation method. We performed simulation studies for a variety of model settings and present our findings with the use of a couple of illustrative examples.

4.4.1 Imposing constraints

The first example concerns the on/off-seq-2 process with parameters $q_{\text{on}} = 0.1$, $q_{\text{off}} = 0.2$, $\lambda_1 = 2$, $\lambda_2 = 1$ and $\mu = 0$. This means that we start with a model in which only births occur and no deaths, and we consider μ as a known parameter. Hence, in this example $\theta = (q_{\text{on}}, q_{\text{off}}, \lambda_1, \lambda_2)^\top$. The size of the data set was fixed, with $n = 120$ and $N = 375$. The results of a simulation study with $B = 1000$ and $b = 10$ are presented in Table 4.1 and Figure 4.2. Table 4.1 shows, for each parameter, the sample mean of the 1000 estimates and the corresponding sample standard deviation. We observe that the sample means for q_{off}, λ_1 and λ_2 do not match with the true parameter values, and the corresponding standard deviations are substantial. This is also reflected in Figure 4.2, which shows, for each parameter, the histogram of the 1000 estimates. The histograms for q_{off}, λ_1 and λ_2 clearly consist of two peaks. The estimates corresponding to one parameter vector θ are displayed in one color, either blue or red, depending on the peak in which the estimate for q_{off} belongs. It shows that there is a one-to-one relation between peaks of the different parameters. Whenever the estimate for q_{off} lies in the lower peak (red), the estimate for λ_1 lies in the lower peak and the estimate for λ_2 lies in the higher peak, and the other way around (blue). We observe that the peaks correspond approximately to the two parameter vectors $\theta_1 = (0.1, 0.1, 1, 2)^\top$ (red), and $\theta_2 = (0.1, 0.2, 2, 1)^\top$ (blue). Note that the blue peaks correspond to the true parameter values of this setting.

By means of further analysis of the on/off-seq-2 process, we can explain why we find two peaks in Figure 4.2. The main reason is that the parameter vectors θ_1 and θ_2 lead to two stochastic processes that are hard to distinguish. This becomes clear by analyzing the distribution of the inter-birth times, the times between consecutive births. Note that these times are i.i.d. We denote the corresponding random variable by T. The time

	q_{on}	q_{off}	λ_1	λ_2
mean	0.1066	0.1625	1.7079	1.3115
sd	0.0036	0.0480	0.4780	0.5315

Table 4.1: Mean values of 1000 estimates, with corresponding standard deviations. On/off-seq-2 process with true parameter values: $q_{\text{on}} = 0.1, q_{\text{off}} = 0.2, \lambda_1 = 2, \lambda_2 = 1$.

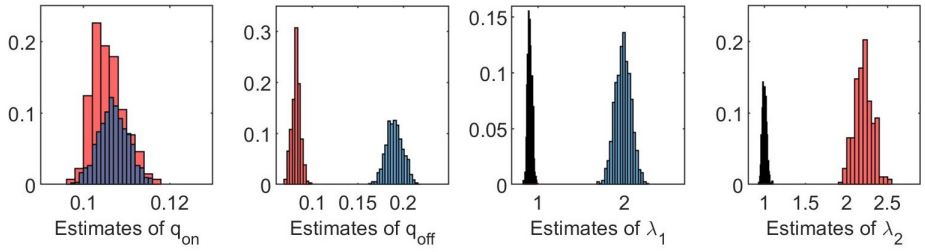

Figure 4.2: Histograms of 1000 estimates. On/off-seq-2 process with true parameter values: $q_{\text{on}} = 0.1, q_{\text{off}} = 0.2, \lambda_1 = 2, \lambda_2 = 1$.

between two births always starts in the on-state, and consists of the time it takes to go back and forth between the on- and off-state, and the time it takes to go through the sequential exponential birth phases. Let $G \in \{1, 2, \dots\}$ be a geometrically distributed random variable with parameter $p = \lambda_1/(\lambda_1 + q_{\text{off}})$, such that $G - 1$ can be interpreted as the number of on/off loops of which the inter-birth time T consist. Then T can be written as the geometric sum

$$T = \sum_{i=0}^{G-1} A_i + \tilde{A}, \tag{4.3}$$

where $A_0 = 0$, the A_i, for $i \geq 1$, are independent and identically distributed as the sum of two exponential random variables with rates $\lambda_1 + q_{\text{off}}$ and q_{on}, and $\tilde{A}$ is distributed as the sum of two exponential random variables with rates $\lambda_1 + q_{\text{off}}$ and λ_2.

Using expression (4.3) for T, we can study its distribution, starting with the expectation and variance of T. Using Wald's equation on the geometric sum, we see that

$$\begin{aligned}
\mathbb{E}[T] &= \mathbb{E}[G-1]\,\mathbb{E}[A_1] + \mathbb{E}[\tilde{A}] \\
&= \left(\frac{q_{\text{off}} + \lambda_1}{\lambda_1} - 1\right) \cdot \left(\frac{1}{q_{\text{off}} + \lambda_1} + \frac{1}{q_{\text{on}}}\right) + \left(\frac{1}{q_{\text{off}} + \lambda_1} + \frac{1}{\lambda_2}\right) \\
&= \frac{1}{\lambda_1} + \frac{1}{\lambda_2} + \frac{q_{\text{off}}}{q_{\text{on}} \cdot \lambda_1}.
\end{aligned}$$

Similarly, with Wald's equation for the variance, we find

$$\begin{aligned}\mathbb{V}\mathrm{ar}[T] &= \mathbb{E}[G-1]\,\mathbb{V}\mathrm{ar}[A_1] + \mathbb{E}[A_1]^2\,\mathbb{V}\mathrm{ar}[G-1] + \mathbb{V}\mathrm{ar}[\tilde{A}] \\ &= \left(\frac{q_{\text{off}}+\lambda_1}{\lambda_1} - 1\right)\cdot\left(\frac{1}{(q_{\text{off}}+\lambda_1)^2} + \frac{1}{q_{\text{on}}^2}\right) \\ &\quad + \left(\frac{1}{q_{\text{off}}+\lambda_1} + \frac{1}{q_{\text{on}}}\right)^2 \cdot \left(\frac{q_{\text{off}}(q_{\text{off}}+\lambda_1)}{\lambda_1^2}\right) + \frac{1}{(q_{\text{off}}+\lambda_1)^2} + \frac{1}{\lambda_2^2} \\ &= \frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2} + \frac{2q_{\text{off}}\lambda_1 + q_{\text{off}}^2 + 2q_{\text{on}}q_{\text{off}}}{\lambda_1^2 q_{\text{on}}^2}.\end{aligned}$$

Interestingly, when computing the expectation and standard deviation of T for the earlier defined parameter vectors θ_1 and θ_2, we observe almost no difference. Parameter θ_1 gives expectation 2.5 with standard deviation 4.92 and parameter θ_2 gives expectation 2.5 with standard deviation 4.82. This means that, for sample sizes of a realistic size, the distribution of T will be indistinguishable for both parameter vectors. This is confirmed by simulations of the distribution of T. For both θ_1 and θ_2, $B = 1000$ realizations of the inter-birth time T were simulated according to (4.3). Figure 4.3 shows the corresponding empirical distribution functions for θ_1 in red, and θ_2 in blue. We see that the distribution functions are almost identical, which explains why the two parameter settings θ_1 and θ_2 are indistinguishable, and two peaks appear in Figure 4.2.

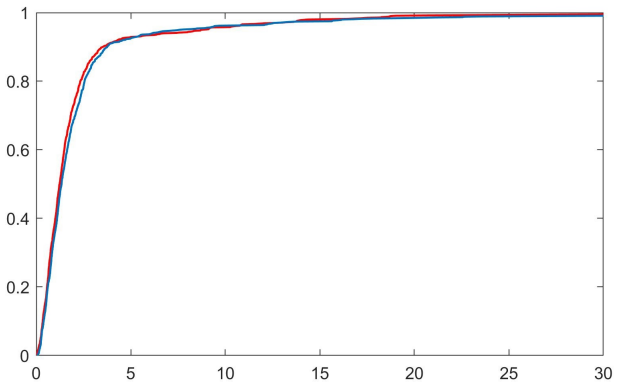

Figure 4.3: Empirical distribution function of T based on 1000 simulated realizations of T for parameter vectors θ_1 (red) and θ_2 (blue).

Intuitively, we can also understand why θ_1 and θ_2 virtually lead to the same stochastic process. Note that in our true parameter setting θ_2, the values for q_{on} and q_{off} are relatively small compared with the values for λ_1 and λ_2, hence the phase process dominates the on/off switch. Because of this timescale separation, the time spent in the off-state between two consecutive births is negligible, and the inter-birth time mainly consist of the two exponential phases with parameters λ_1 and λ_2. Interchanging the two phases will therefore have a modest effect on the inter-birth times, as long as the probability of jumping from

state $X_t = 1$ to $X_t = 2$ stays the same. This probability is equal to $\lambda_1/(\lambda_1 + q_{\text{off}})$, hence if q_{off} is adjusted in the right way, the new situation virtually yields the same stochastic process. This is exactly what describes the difference between θ_2 and θ_1. The parameter values for λ_1 and λ_2 are swapped, and the probability $\lambda_1/(\lambda_1 + q_{\text{off}}) = 10/11$ in both situations.

We conclude that in some parameter settings, the shape of the likelihood function is such that numerical maximization can lead to multiple estimates of θ. A way to overcome this numerical complication is by imposing constraints when maximizing the likelihood function. Table 4.2 and Figure 4.4 show the results of a simulation study equal to the one above, with the only difference that the likelihood functions are maximized under the constraint $\lambda_1 \geq \lambda_2$, making it no longer possible to interchange λ_1 and λ_2. We see from Table 4.2 that the mean values of the 1000 estimates lie close to the true parameter values, and that the standard deviations for the last three parameters decreased considerably. Figure 4.4 shows us that the histograms of all parameters only have one peak now that we imposed the constraint on λ_1 and λ_2.

	q_{on}	q_{off}	λ_1	λ_2
mean	0.1066	0.1911	1.9910	1.0004
sd	0.0036	0.0096	0.0960	0.0278

Table 4.2: Mean values of 1000 estimates, with corresponding standard deviations, obtained under the constraint $\lambda_2 \geq \lambda_1$. On/off-seq-2 process with true parameter values: $q_{\text{on}} = 0.1, q_{\text{off}} = 0.2, \lambda_1 = 2, \lambda_2 = 1$.

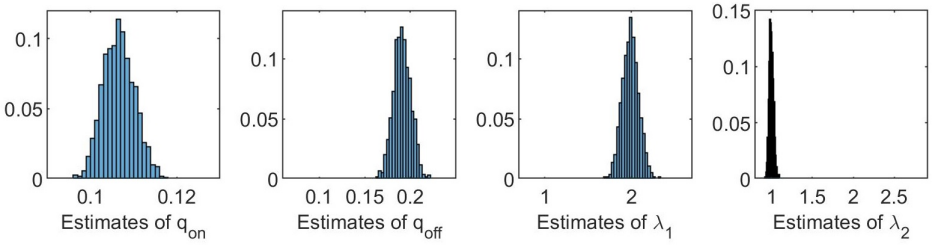

Figure 4.4: Histograms of 1000 estimates obtained under the constraint $\lambda_2 \geq \lambda_1$. On/off-seq-2 process with true parameter values: $q_{\text{on}} = 0.1, q_{\text{off}} = 0.2, \lambda_1 = 2, \lambda_2 = 1$.

4.4.2 The influence of n and N

In this section we investigate the influence of n and N on the accuracy of the estimation method. To illustrate our findings, we use the example as above, hence $q_{\text{on}} = 0.1$, $q_{\text{off}} = 0.2$, $\lambda_1 = 2$, $\lambda_2 = 1$, with the small adjustment that the death rate of the simulated data, μ, now equals 0.3. Hence, we analyze a model in which both births and deaths occur, and of which the death rate μ is an unknown parameter as well. Note that the distribution

of T does not depend on the value of μ, hence we again need to impose the constraint $\lambda_1 \geq \lambda_2$ when maximizing the likelihood function.

To investigate the influence of n on the accuracy of the estimation method, we performed simulations for increasing values of n with $N = 350$ fixed. We chose $n = 50$, $n = 100$, $n = 200$, $n = 500$ and $n = 1000$. The results for $B = 1000$ and $b = 10$ are shown in Table 4.3 and Figures 4.5–4.9. In a few cases, the estimate $\hat{\theta}$ ended up at the boundary of the domain $\mathcal{D}$ over which the likelihood function was maximized. This numerical issue was easily solved by enlarging the domain, after which the estimate ended up in the interior of $\mathcal{D}$. Table 4.3 shows, for the increasing values of n, the sample mean of the 1000 estimates, with the sample standard deviation between brackets. We see that, for all five parameters, the sample means lie closer to the true parameter values as n increases. Furthermore, the standard deviations decrease as n increases. This is also seen in Figures 4.5–4.9, which show for each parameter the histograms of the 1000 estimates for the increasing values of n. In each figure, the limits of the x-axis are equal for the five histograms, which makes it immediately visible that the histograms become narrower when n increases.

n	q_{on}	q_{off}	λ_1
50	0.1151 (0.0069)	0.1732 (0.0249)	1.9214 (0.2848)
100	0.1072 (0.0045)	0.1847 (0.0190)	1.9607 (0.2063)
200	0.1035 (0.0032)	0.1911 (0.0133)	1.9756 (0.1455)
500	0.1015 (0.0018)	0.1967 (0.0091)	1.9913 (0.0934)
1000	0.1007 (0.0013)	0.1979 (0.0063)	1.9924 (0.0635)

n	λ_2	μ
50	1.0311 (0.1049)	0.3009 (0.0057)
100	1.0132 (0.0717)	0.3005 (0.0039)
200	1.0082 (0.0468)	0.3004 (0.0028)
500	1.0035 (0.0284)	0.3002 (0.0018)
1000	1.0023 (0.0194)	0.3001 (0.0013)

Table 4.3: Mean values of 1000 estimates for increasing values of n and $N = 350$, with corresponding standard deviation between brackets. On/off-seq-2 process with true parameter values: $q_{\text{on}} = 0.1, q_{\text{off}} = 0.2, \lambda_1 = 2, \lambda_2 = 1, \mu = 0.3$.

We have seen that the accuracy of the estimation method can be increased by choosing a higher value of n. However, in practical situations it is not always possible to increase n. This is, for example, the case in the application studied in Section 4.5. One experiment measures the number of mRNA molecules in a single cell over time, but the lifetime of a cell is limited. The number of experiments N, however, *can* be increased. To investigate the influence of N on the accuracy of the estimation method, we performed simulations for increasing values of N with $n = 100$ fixed. We considered $N = 200$, $N = 350$, $N = 500$, $N = 750$ and $N = 1000$. The results for $B = 1000$ and $b = 10$ are given in Table 4.4. For each value of N, this table shows again the sample mean of the 1000 estimates with

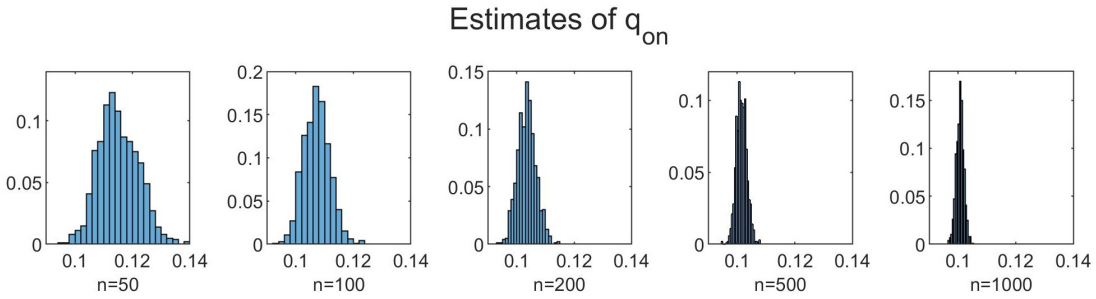

Figure 4.5: Histograms of the obtained estimates of q_{on} for increasing values of n.

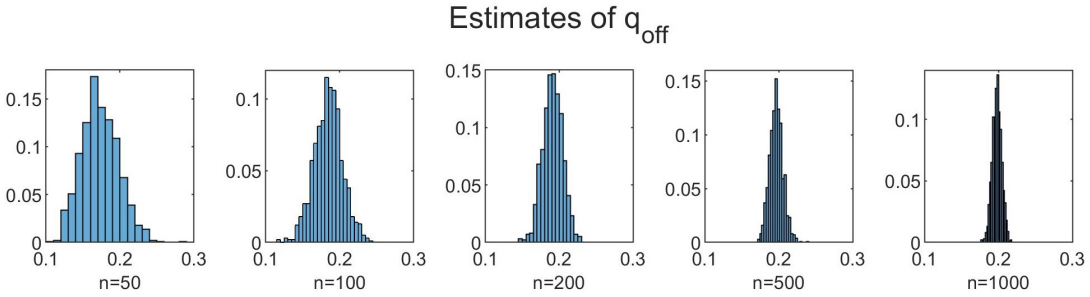

Figure 4.6: Histograms of the obtained estimates of q_{off} for increasing values of n.

the sample standard deviation between brackets. We see that for each parameter, the mean values lie close to the true parameter value, but do not improve as N increases. This means that the bias of the estimates is mainly determined by the value of n, which is related to how much information is given by one experiment. However, Table 4.4 also shows that the standard deviations do decrease as N increases, and in this way provides insight in how the accuracy increases as a function of N.

4.4.3 On/off-seq-3 process

In the first part of the numerical study, we have analyzed the on/off-seq-2 process. In this section we explore the numerical complications related to the likelihood maximization for the on/off-seq-L process with $L > 2$, and we investigate the accuracy of the estimation method for the on/off-seq-3 process. First note that for $L > 2$, the model is partially unidentifiable, since interchanging the parameters $\lambda_2, \ldots, \lambda_L$ yields an identically distributed process $\{M_t\}$. Hence, when performing likelihood maximization, a fixed order of these parameters should be chosen.

The analysis on the inter-birth times of the on/off-seq-2 process can be extended for $L > 2$. The inter-birth time T can still be written as the geometric sum in (4.3), but $\tilde{A}$ is now distributed as the sum of L exponential random variables with rates $\lambda_1 + q_{\text{off}}$, $\lambda_2, \ldots, \lambda_L$. This means that $\mathbb{E}[T]$ and $\mathbb{V}\text{ar}[T]$ only change by factors $\frac{1}{\lambda_3} + \cdots + \frac{1}{\lambda_L}$ and

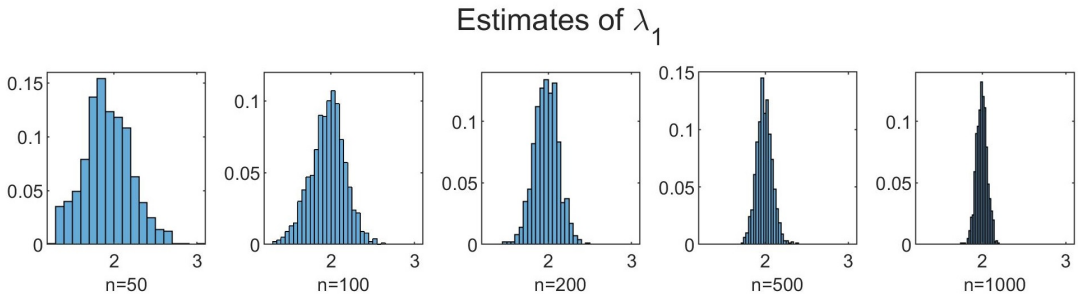

Figure 4.7: Histograms of the obtained estimates of λ_1 for increasing values of n.

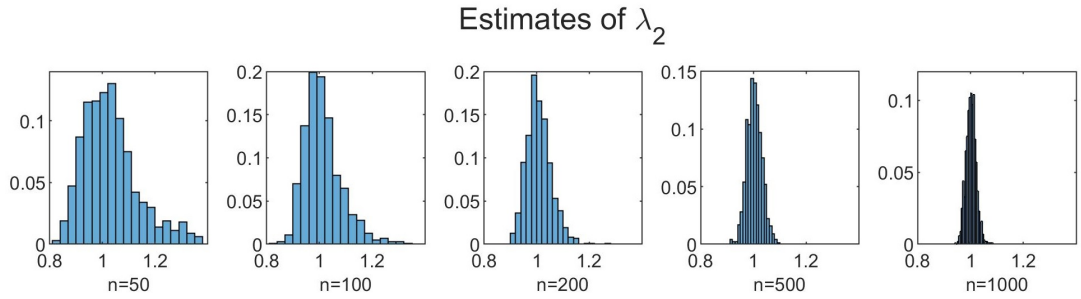

Figure 4.8: Histograms of the obtained estimates of λ_2 for increasing values of n.

$\frac{1}{\lambda_3^2} + \cdots + \frac{1}{\lambda_L^2}$, respectively. We have

$$\mathbb{E}[T] = \frac{1}{\lambda_1} + \cdots + \frac{1}{\lambda_L} + \frac{q_{\text{off}}}{q_{\text{on}} \cdot \lambda_1}.$$

Similarly, with Wald's equation for the variance, we find

$$\mathbb{V}\text{ar}[T] = \frac{1}{\lambda_1^2} + \cdots + \frac{1}{\lambda_L^2} + \frac{2q_{\text{off}}\lambda_1 + q_{\text{off}}^2 + 2q_{\text{on}}q_{\text{off}}}{\lambda_1^2 q_{\text{on}}^2}.$$

This means that the same reasoning holds as for the on/off-seq-2 process, and additional constraints on λ_1 with respect to $\lambda_2, \ldots, \lambda_L$ are needed to make sure that the likelihood function has a unique maximum.

To investigate the accuracy of the estimation method for the on/off-seq-3 process, we performed a variety of simulation studies. We present our findings by means of two different examples. The first example is the on/off-seq-3 process with parameters $q_{\text{on}} = 0.2$, $q_{\text{off}} = 0.5$, $\lambda_1 = 0.5$, $\lambda_2 = 2$, $\lambda_3 = 4$ and $\mu = 0.1$. Table 4.5 and Figure 4.10 show the simulation results for this example under the constraint $\lambda_1 \leq \lambda_2 \leq \lambda_3$, with $B = 1000$, $b = 10$ and data size $n = 1000$, $N = 350$. Table 4.5 shows, for each parameter, the sample mean and corresponding sample standard deviation of the 1000 estimates. We see that the mean values for parameters q_{on}, λ_2, λ_3 and μ lie close to the true parameter

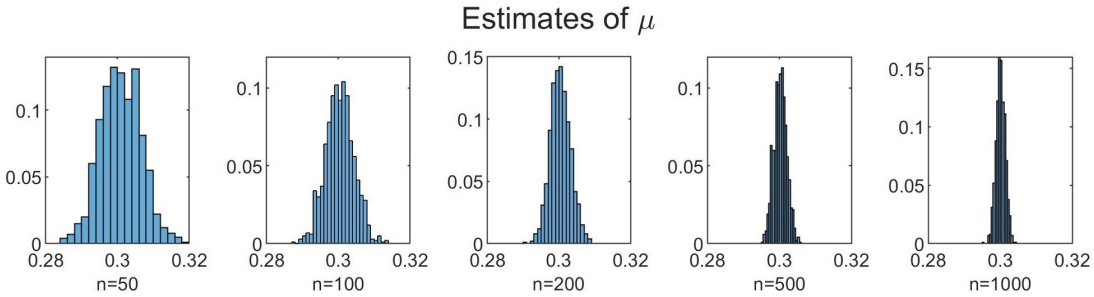

Figure 4.9: Histograms of the obtained estimates of μ for increasing values of n.

N	q_{on}	q_{off}	λ_1
200	0.1072 (0.0059)	0.1849 (0.0252)	1.9642 (0.2679)
350	0.1072 (0.0045)	0.1847 (0.0190)	1.9607 (0.2063)
500	0.1071 (0.0038)	0.1848 (0.0151)	1.9639 (0.1701)
750	0.1072 (0.0031)	0.1850 (0.0124)	1.9627 (0.1376)
1000	0.1072 (0.0027)	0.1849 (0.0106)	1.9609 (0.1176)

N	λ_2	μ
200	1.0199 (0.0971)	0.3005 (0.0054)
350	1.0132 (0.0717)	0.3005 (0.0039)
500	1.0097 (0.0577)	0.3006 (0.0032)
750	1.0082 (0.0459)	0.3007 (0.0027)
1000	1.0078 (0.0384)	0.3007 (0.0023)

Table 4.4: Mean values of 1000 estimates for increasing values of N and $n = 100$, with corresponding standard deviation between brackets. On/off-seq-2 process with true parameter values: $q_{\text{on}} = 0.1, q_{\text{off}} = 0.2, \lambda_1 = 2, \lambda_2 = 1, \mu = 0.3$.

values. The mean values for parameters q_{off} and λ_1, however, exceed the true parameter values. This is also visible in Figure 4.10, which shows for each parameter the histogram of the 1000 estimates. The histograms for q_{off} and λ_1 show some outliers which increase the corresponding means. This example confirms that when L increases it becomes more difficult to accurately estimate all model parameters from the data. This is supported by the fact that the variance of T increases when L grows. Hence, as to be expected, for larger L more data is needed (i.e. by increasing n) to obtain a similar accuracy as for models with a smaller L.

For some applications it may be more realistic to assume that all $\lambda_i, i = 1, \ldots, L$, are equal. Under this assumption, the accuracy of the estimation method may increase substantially. We illustrate this by the second example. We consider the on/off-seq-3 process with parameters $q_{\text{on}} = 0.25$, $q_{\text{off}} = 1$, $\lambda_1 = \lambda_2 = \lambda_3 = \lambda = 10$ and $\mu = 2$, hence $\theta = (q_{\text{on}}, q_{\text{off}}, \lambda, \mu)^\top$. The results of a simulation study with $B = 1000$, $b = 50$, $n = 120$ and $N = 375$ are presented in Table 4.6 and Figure 4.11. Table 4.6 shows, for each

	q_{on}	q_{off}	λ_1	λ_2	λ_3	μ
mean	0.2008	0.5441	0.5360	1.9895	3.9995	0.1000
sd	0.0037	0.1463	0.1139	0.4370	0.7865	0.0006

Table 4.5: Mean values of 1000 estimates, with corresponding standard deviations. True parameter values: $q_{\text{on}} = 0.2$, $q_{\text{off}} = 0.5$, $\lambda_1 = 0.5$, $\lambda_2 = 2$, $\lambda_3 = 4$, $\mu = 0.1$.

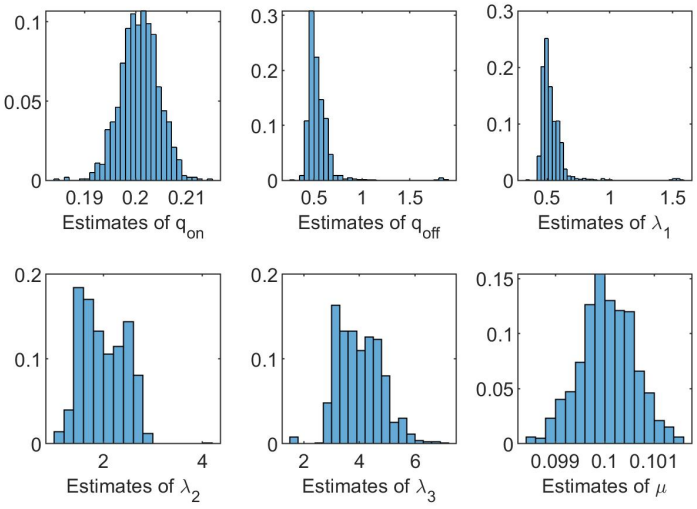

Figure 4.10: Histograms of 1000 estimates. On/off-seq-3 process with true parameter values: $q_{\text{on}} = 0.2$, $q_{\text{off}} = 0.5$, $\lambda_1 = 0.5$, $\lambda_2 = 2$, $\lambda_3 = 4$, $\mu = 0.1$.

parameter, the sample mean and corresponding sample standard deviation of the 1000 estimates. We see that the mean values of the parameters are close to the true parameter values. This is reflected in Figure 4.11, which shows for each parameter the histogram of the 1000 estimates. The histograms are nicely shaped around the true parameter values. Note that the size of the data in this example is substantially smaller than in the previous example.

	q_{on}	q_{off}	λ	μ
mean	0.2547	0.9727	10.1153	2.0282
sd	0.0049	0.0253	0.2028	0.0451

Table 4.6: Mean values of 1000 estimates, with corresponding standard deviations. True parameter values: $q_{\text{on}} = 0.25, q_{\text{off}} = 1, \lambda = 10$ and $\mu = 2$.

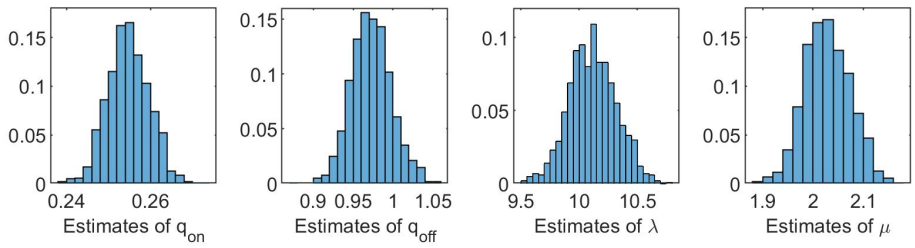

Figure 4.11: Histograms of 1000 estimates. On/off-seq-3 process with true parameter values: $q_{\text{on}} = 0.25, q_{\text{off}} = 1, \lambda = 10$ and $\mu = 2$.

4.5 mRNA transcription

In this section we apply the estimation method for the on/off-seq-L process, as described at the end of Section 4.3, to real data of mRNA counts. We first describe in detail the biological process of mRNA transcription, and then describe a model selection procedure that we performed on the data with respect to various on/off-seq-L processes.

4.5.1 Biological background

Proteins play a major role in the structure and functioning of cells. In fact, all physiological processes in cells depend on proteins. The information needed for the synthesis of proteins is stored in the DNA; think of it as a collection of recipes. Specific parts of the DNA, called genes, contain the information for a particular protein, and can be seen as one recipe. When a protein is needed, the information in the corresponding gene is used for the synthesis of this protein in a process called *gene expression*. Gene expression takes place in two steps, see Figure 4.12. In the first step, called *transcription*, the information in the gene is copied into an mRNA molecule. In the second step, called *translation*, the copied information in the mRNA molecule is used to make the corresponding protein. By transcription, multiple identical mRNA molecules can be produced from one gene, and by translation each of these mRNA molecules can produce multiple identical proteins. In this way, the proteins can be synthesized with their own efficiency according to the needs of the cell, despite the fact that each cell contains only one or two copies of a specific gene. Interestingly, gene expression is constructed in this way in all cells, from bacteria to humans. We focus on the transcription step in gene expression. It is known that in bacteria the stochasticity in gene expression stems largely from transcription [36], which is why a stochastic model for this process is appropriate.

Figure 4.12: Steps of protein synthesis.

The transcription of mRNA molecules is a complex process. After the transcription

of an mRNA molecule has been initiated, it takes multiple sequential phases before the molecule is eventually produced. Biologically, mRNA transcription takes place through the following steps: first, the molecule RNA polymerase binds to the DNA and slides along the DNA to find a transcription start site, called promoter. Once it has found a start site it binds firmly and the transcription begins. The RNA polymerase moves along the gene while copying the genetical code step by step. Once it reaches the stop site, it releases itself and the new mRNA transcript from the DNA. From there, the process can be repeated to produce more mRNA molecules. The mRNA transcription can be controlled by a process called *gene repression*. The promoter can bind to repressors for a period of time in which RNA polymerase cannot reach the start site to initiate transcription. This causes the promoter to switch between an active state, free from repressors, and an inactive state, bound by repressors.

The on/off-seq-L process has been found to be a realistic model for mRNA transcription [51, 30], and combines the active/inactive switch of the promoter with the sequential phases of transcription. The phases in the transcription process that contribute to the transcription rate the most are called *rate limiting*, and differ per promoter. Phases that are relatively fast compared to other phases generally do not need to be included in the model. Likewise, it depends on the promoter whether or not the active/inactive mechanism has a (substantial) effect on the transcription dynamics. If the time spent in the inactive state is relatively short compared to the time spent in the active state, it could be decided not to include an on/off mechanism in the model. The model that leads to the best representation of the transcription process can be identified either based on biological considerations or by means of a statistical model selection procedure.

4.5.2 Model selection

In this section we describe a model selection procedure that we performed for mRNA data corresponding to the so-called λ RM promoter [30], which were kindly provided by prof. A.S. Ribeiro from Tampere University, Finland. The available data set consists of measurements on the number of mRNA molecules in a total of 775 single cells, hence $N = 775$. Each cell was measured every minute over a period of two hours, hence $\Delta = 1$ and $n = 121$. We used the on/off-seq-L process to describe the data and applied the Erlangization method as described in Section 4.3 to evaluate the likelihood function and obtain maximum likelihood estimates. We performed our model selection on six different models, arising from the combination of whether or not there is an on/off mechanism, and if the birth process consists of 1, 2 or 3 phases. This means that next to the on/off-seq-1, on/off-seq-2 and on/off-seq-3 models, we considered the seq-1, seq-2 and seq-3 models in which the on/off mechanism is omitted. In line with [30], we assumed that $\lambda_1 \leq \lambda_2 \leq \lambda_3$.

The results of the model selection, with $b = 10$, are shown in Table 4.7. This table shows for each model the maximum likelihood estimates of the parameters in the first five columns, the sixth column presents the corresponding likelihood values, and the Akaike information criterion (AIC) is shown in the last column. We see that the model that leads to the best fit should contain an on/off mechanism, since the lowest AIC values are found for these models. Because these values are close to each other, we computed the

	q_{on}	q_{off}	λ_1	λ_2	λ_3	logL	AIC
seq-1	-	-	0.0144	-	-	-3569.9	7141.8
seq-2	-	-	0.0144	8.9245	-	-3569.4	7142.7
seq-3	-	-	0.0144	9.9536	9.9875	-3569.3	7144.6
on/off-seq-1	0.0249	0.0608	0.0496	-	-	-3475.2	6956.5
on/off-seq-2	0.0303	0.4089	0.2220	0.2221	-	-3474.8	6957.6
on/off-seq-3	0.0312	0.4410	0.2314	0.2314	9.9983	-3475.6	6961.3

Table 4.7: Model selection for the λ RM promoter data with $\lambda_1 \leq \lambda_2 \leq \lambda_3$. The columns show the maximum likelihood estimates, the loglikelihood values, and the AICs, respectively.

relative likelihood values to give more insight in the differences. Let $\hat{\mathcal{L}}_i$ be the maximum likelihood value for the on/off-seq-i model corresponding to the results in Table 4.7. Then the relative likelihood, $\hat{\mathcal{L}}_2/\hat{\mathcal{L}}_1$, is equal to 0.5769, and the relative likelihood, $\hat{\mathcal{L}}_3/\hat{\mathcal{L}}_1$, is equal to 0.0907. These values are not close to 1, which indicates that the on/off-seq-1 model seems to be the best model for this type of data, in line with the findings of [30]. Note that the AIC-based conclusion that a model with three phases is least suitable for the data is supported by the fact that the maximum likelihood estimates of λ_3 are relatively large with respect to the other parameters. Then comparing the maximum likelihood estimates for the on/off-seq-1 model in Table 4.7 with the results in [30], we see that the estimates are of the same order of magnitude, but do not match precisely. This can be explained by the fact that in [30], the likelihood function is computed from observations of the transcription intervals and not from the mRNA counts. As mentioned in the introduction of this chapter, these intervals are not observed precisely and therefore censoring is needed to compute the likelihood function.

4.6 Discussion

Motivated by a biological application, we have studied the on/off-seq-L process, a BD process with births occurring according to a sequential process consisting of multiple phases and regulated by an on/off mechanism. We have mathematically defined the on/off-seq-L process and have shown that it can be seen as a QBD process. The latter enables the use of the Erlangization technique as introduced in Chapter 3 to approximate the likelihood function. Maximum likelihood estimates can then be obtained by numerical optimization of this likelihood.

In a numerical study, we have investigated the accuracy of this estimation method for the on/off-seq-L process, and have explored numerical complications related to the likelihood maximization. We have shown that for some parameter settings the shape of the likelihood function is such that numerical maximization can lead to multiple estimates of θ. It is therefore necessary to impose constraints on the order of $\lambda_1, \ldots, \lambda_L$ when maximizing the likelihood function. Under these constraints, the estimation method works

as expected. We have seen that the estimation method yields accurate results, and that the accuracy improves as n or N increases. As illustrated for $L = 3$, the estimation method can also be applied for processes with $L > 2$, but more observations are needed to obtain a similar accuracy as for $L = 2$.

We note that the results that we obtained hold for a parameter setting where the phase process dominates the on/off switch. That is, the values for q_{on} and q_{off} are relatively small compared to the values for $\lambda_1, \ldots, \lambda_L$. However, parameter settings for which this is not the case should also be explored. Recall that the random variable $G-1$, as in the definition of T (4.3), can be seen as the number of on/off loops of which the inter-birth time consists. Furthermore, $\mathbb{E}[G-1] = q_{\text{off}}/\lambda_1$, hence the ratio of these two parameters play a major role in how the process behaves. We suspect that there are three different regimes that need to be distinguished with respect to the timescales of the parameters:

- λ_1 is substantially higher than q_{off}. In this case $\mathbb{E}[G-1]$ is small and the phase process dominates the on/off switch. This regime corresponds to the settings studied in Section 4.4.

- λ_1 is substantially smaller than q_{off}. In this case $\mathbb{E}[G-1]$ is large and the on/off switch dominates the phase process. In view of performing statistical inference on the model, this does not seem to be a relevant regime in any practical situation. Only very view births will occur and therefore the on/off mechanism will not be detectable from data on the population size.

- Both λ_1 and q_{off} are of the same order of magnitude. In view of performing statistical inference on the model, this seems to be a relevant regime when $\mathbb{E}[G-1] \leq c$, for some constant c small enough. At the same time, we expect it to be a complicated regime with its own numerical complications. Preliminary simulation studies suggest that, unless n is large, the value of c will be hard to distinguish from the data, and hence the corresponding parameters are hard to estimate.

The possible regimes leads us to an important direction for further research. It is interesting to investigate whether there are more relevant regimes and how this can be confirmed mathematically. Moreover, the parameter estimation method should be explored for the last regime, in which all parameters are of the same order of magnitude. Here, one of the questions is whether it is possible to find constraints on the model parameters under which the likelihood maximization will result in accurate estimates.

5. Multivariate population processes

We consider discrete-time multivariate population processes under Markov modulation. Our objective is to estimate the model parameters, based on periodic observations of the network population vector. These parameters relate to the arrival, routing and departure processes, but also to the (unobservable) Markovian background process. When opting for the classical likelihood-based approach, the evaluation of the likelihood is problematic. We show however, how an accurate saddlepoint approximation can be used. Numerical experiments illustrate our method and show that even under relatively complicated conditions the parameters are estimated relatively precisely.

5.1 Introduction

Population processes are stochastic processes that record the dynamics of the number of individuals in a population. Owing to their widespread use in for instance biology, ecology, and chemical reaction networks, they have become a key object of study in statistics and applied probability. In its simplest form a population process describes the fluctuations of the population size at a single location. Many practically relevant situations, however, correspond to considerably more general settings. In the first place, the population process often lives on a *multi-node* (rather than single-node) network. This means that individuals can enter and leave the nodes of the network, but in addition they can move between its nodes. Secondly, in many situations the dynamics of the population are affected by exogenous, often unobservable, factors; think of temperature affecting the spread of bacteria or weather conditions affecting the mobility of the individuals. In these cases it is desirable to add an underlying modulating process to the model, referred to as the *background process*.

Due to the ubiquity of multivariate modulated population processes across a wide range of scientific disciplines, there is a clear need for sound statistical techniques to estimate the underlying parameters. In this chapter we devise such a method based on observations of the network population vector. We do so in a discrete-time context, with the background process corresponding to a finite state-space Markov chain. This means that we are in the context of *Markov modulation*, with the values of the parameters pertaining to the arrival, routing, and departure processes being a function of the state of the background process.

In the setting considered, parameter estimation can be seen as a highly challenging inverse problem. When developing an estimation procedure, one needs to cope with two major intrinsic complications.

- In the first place, as we have access to the network population vector only, we do not observe the number of arrivals, the number of individuals that are routed between each of the node pairs, and the number of departures, but only the *net effect* of these processes. This effectively means that in general we cannot trace how individuals have moved through the network.

- The second complication is that we assume that we cannot observe the background process (making its state a hidden variable). The challenge is to infer from the observations the parameters of the Markovian background process, and the (background-state dependent) parameters pertaining to the arrival, routing, and departure processes.

There is a considerable body of work on inverse problems for continuous-time population processes. In the first place we refer to for example [24, 67, 18, 19, 70] for parameter estimation procedures for univariate birth-death processes without modulation. In these papers the case is considered where the population is observed at discrete times only, hence the individual births and deaths are not observed directly. In addition there are various papers on estimation techniques for infinite-server queues (which can be seen as population processes in which the times the individuals spend in the system are independent of each other) without modulation. In this context we mention [55], in which the service-time distribution is estimated without direct observations of the service times, and [11], which treats the estimation of the arrival rate and the service-time distribution from observations of the population size. A separate branch of the literature focuses on parameter estimation for stochastic processes with a Markovian, unobserved background process. In this respect we mention [31, 32], which concentrate on the class of Markovian binary trees and continuous-time observations or demographic data. In addition, when focusing on a Markovian arrival process only, rather than the resulting population process, in [15, 50] estimation procedures based on discrete-time observations are presented. We finally refer to Chapter 2, in which a parameter estimation procedure for a univariate population process under Markov modulation is proposed and assessed, based on discrete-time observations of the population size.

The work presented in this chapter concerns parameter estimation for a multivariate population process, and can as such be seen as part of the broader area of network science. There is a strong relation with the subdiscipline that focuses on the statistical analysis of network data. We refer to [41, Chapters 8 and 9] for more background on statistical procedures for stochastic processes on networks. It is noted, though, that existing theory predominantly concentrates on situations in which the routing process on the network—often referred to as the network flow—is fully observed, which contrasts with the situation considered in this chapter.

Importantly, to the best of our knowledge, there are no procedures available for estimating the parameters of modulated multivariate population process, based on observations of the network population vector. One could pursue an approach based on maximum

likelihood, but evaluating the likelihood is generally problematic. The main difficulty lies in the complexity of the model, in terms of the size of the underlying network and the fact that there is a modulating background process. As a consequence, typically no closed-form expression for the likelihood can be given; in addition, in the special cases where it is possible to obtain such an explicit expression, there are often numerical complications. We therefore take another approach, which combines the following two ideas:

- Due to the structure of the model, it is possible to set up a procedure to compute for each point in time the joint moment generating function (mgf) pertaining to the network population vector.

- We then apply the technique of *saddlepoint approximation* to compute an approximation of the likelihood, and maximize this approximation over the unknown parameters. The saddlepoint approximation provides a (typically highly accurate) approximation of the probability mass function of a random vector, based on the corresponding joint mgf.

The saddlepoint technique has been developed in the 1950s by Daniels [22]; for a textbook treatment see e.g. [17]. For specific models closed-form expressions for saddlepoint approximations have been obtained. In this respect we refer to [1] for explicit approximations of the transition densities and cumulative distribution functions of Markov processes, whereas in [23] general birth processes are considered. The references [10, 60] provide extensive general accounts of the use of saddlepoint techniques in statistics. A few papers where saddlepoint expansions have been used to approximate the likelihood are [54] which considers the context of the INAR(p) model, [26] where the focus is on the distribution of the sum of independent non-identically distributed binomial random variables, and [24] which aims at estimating the birth and death rates of a linear birth-and-death process.

We proceed by discussing this chapter's main contributions in more detail. First and foremost, to our best knowledge, we are the first to develop a parameter estimation procedure in the highly general and comprehensive setup of a multivariate population process under Markov modulation, based on periodic observations of the network population vector. Our approach is likelihood-based, but only in special (small) networks the likelihood can be computed in closed form, which is why we approximate the likelihood relying on the saddlepoint approximation. A prerequisite for using the saddlepoint technique is the availability of the mgf corresponding to the network population vector at multiple points in time. We present an efficient technique to evaluate this mgf, by computing the mgf of the network population vector at one observation time conditionally on the population vector at the previous observation time. Then this mgf is used to approximate the likelihood, which numerically boils down to solving a convex optimization problem. Subsequently the approximated likelihood is maximized over the parameter space to find approximate values for the maximum likelihood estimates of the model parameters. The last contribution concerns numerical experiments, which assess the performance of our parameter estimation technique. They show that even under relatively complicated conditions (modulation, a multi-node system), following our approach, the parameters can be estimated relatively precisely. The examples involve single- and multi-node networks,

with and without modulation, and illustrate the factors that affect the procedure's performance.

The remainder of this chapter is organized as follows. In Section 5.2 we formally define the multivariate population process under Markov modulation, and we state the estimation problem. Section 5.3 focuses on two examples of small networks (a single-node model and a tandem network of two nodes), showing how in these cases the likelihood can be computed explicitly. This section also points out how the expressions for the likelihood become increasingly involved if the number of network nodes increases. In Section 5.4 we show how the likelihood can be evaluated using saddlepoint approximations; this section also includes the method to compute the mgf of the network population vector. We show how the approximation of the likelihood can be used to estimate the model parameters, and investigate the accuracy of this estimation method by numerical studies in Section 5.5. We conclude the chapter with a discussion in Section 5.6.

5.2 Model and estimation

As mentioned in the introduction, this chapter considers a population process on a network with finitely many nodes. Individuals can arrive at each of the nodes, follow a probabilistic route through the network, and potentially leave the network. We impose Markov modulation: all parameters in the model are driven by a discrete-time Markov chain, where each state corresponds to a different set of parameter values. In this section, we first present a detailed mathematical description of our Markov modulated multivariate population process, and then state the corresponding parameter estimation problem.

We throughout adopt the convention that vectors are printed in bold; we denote by $\boldsymbol{x}(k)$ the k-th entry of the vector $\boldsymbol{x}$. As usual, random variables and matrices are denoted by capital letters. We use $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ to denote the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$ (whose dimensions are then assumed to be compatible). We write $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

5.2.1 The model

We start by introducing the *background process* $\{X_k\}_{k\in\mathbb{N}_0}$. This is an irreducible discrete-time Markov chain with finite state space $E = \{1, \ldots, d\}$, $d \in \{2, 3, \ldots\}$. We define by $P = (p_{ij})_{i,j=1}^{d}$ the corresponding $(d \times d)$ transition probability matrix, $\boldsymbol{\alpha}$ the corresponding initial state distribution (i.e., $\alpha_i := \boldsymbol{\alpha}(i) = \mathbb{P}(X_0 = i)$), and $\boldsymbol{\pi}$ the (unique) stationary distribution. Recall that $\boldsymbol{\pi}^\top P = \boldsymbol{\pi}^\top$. The background process modulates the network's dynamics in a way we make precise below.

We study a network with $L \in \mathbb{N}$ nodes on which we define the multivariate population process $\{\boldsymbol{M}_k\}_{k\in\mathbb{N}_0}$, where the vector $\boldsymbol{M}_k$ records the number of individuals present at the L nodes at time k. This population process is the result of an arrival process, a routing mechanism by which individuals jump between the nodes, and a departure process. We now introduce these individual ingredients.

- Denote by $\{\boldsymbol{A}_k\}_{k\in\mathbb{N}}$ the arrival process, where $\boldsymbol{A}_k \in \mathbb{N}^L$ represents a vector that

counts the number of arrivals at each of the L nodes at time k. We assume that these arrivals stem from a parametric class, where the parameters depend on the value of X_{k-1}, i.e., the state of the background process at time $k-1$; the arrival process is thus Markov modulated. More precisely, given $X_{k-1} = i$, for some $i \in E$, the moment generating function (mgf) of the arrivals at time k is assumed to exist and given by (for $\boldsymbol{s} \in \mathbb{R}^L$)

$$\phi_{k,i}(\boldsymbol{s}) := \mathbb{E}[e^{\langle \boldsymbol{s}, \boldsymbol{A}_k \rangle} | \, X_{k-1} = i], \tag{5.1}$$

with the corresponding cumulant generating function (cgf) denoted by $\psi_{k,i}(\boldsymbol{s}) := \log \phi_{k,i}(\boldsymbol{s})$. In the sequel, we let the individual components of $\boldsymbol{A}_k$ be time-homogeneous and independent, and let $\boldsymbol{A}_k(\ell)$ have a Poisson distribution with parameter $\lambda_i^{(\ell)} \geqslant 0$, given $X_{k-1} = i$. In this case

$$\psi_{k,i}(\boldsymbol{s}) \equiv \psi_i(\boldsymbol{s}) = \sum_{\ell=1}^{L} \lambda_i^{(\ell)} (e^{\boldsymbol{s}(\ell)} - 1). \tag{5.2}$$

We emphasize that the use of other choices of the arrival process is straightforward, as long as the mgf defined in (5.1) exists and is known.

- The routing and departure processes are Markov modulated as well. To describe these processes, we first define for each $\ell \in \{1, \ldots, L\}$ the vector-valued process $\{\boldsymbol{D}_k^{(\ell)}\}_{k \in \mathbb{N}}$, where $\boldsymbol{D}_k^{(\ell)} \in \mathbb{N}^{L+1}$. For ℓ' between 1 and L, $\boldsymbol{D}_k^{(\ell)}(\ell')$ counts the number of individual jumps out of node ℓ towards node ℓ' at time k, whereas the $\boldsymbol{D}_k^{(\ell)}(L+1)$ records the number of individuals that leave the network from node ℓ at time k. We say that a jump from a node to itself is the same as staying at the node. Importantly, in our model all individuals can move independently of each other through the network and do not have to wait for each other. Let $r_i^{(\ell,\ell')} \in [0,1]$ be the probability of an individual at node ℓ to jump to node ℓ' at an arbitrary time point when the background state is i. Also,

$$r_i^{(\ell,0)} := 1 - \sum_{\ell=1}^{L} r_i^{(\ell,\ell')}$$

(which is a number in $[0,1]$) denotes the probability of an individual to leave the network from node ℓ at any time point at which the background state is i. If $r_i^{(\ell,0)} = 0$, individuals cannot leave the network from node ℓ when the background process is in state i. Note that for each $k > 0$, given $\boldsymbol{M}_{k-1}$ and X_{k-1}, the vectors $\boldsymbol{D}_k^{(\ell)}$ are independent. In addition, for a given ℓ the vector $\boldsymbol{D}_k^{(\ell)}$ follows a multinomial distribution.

In our model we let the change of the background process happen *after* the arrivals, the routing and the departures. We remark, however, that this choice does not impose any

restriction: in the very same manner we can deal with the analogous model in which the background process jumps before the arrivals, the routing and the departures.

Furthermore, both the routing and the departures occur *before* the arrivals, which implies that newly arrived individuals can only leave the node the next timeslot at the earliest. It is also possible to assume that the arrivals occur before the departures and routing. This leads to a slightly different model, in which individuals who leave the system in the same interval as they arrive are included in both the arrival process and departure process, although they are not visible in the population process $\{M_k\}$.

We proceed by introducing various quantities related to $\{\boldsymbol{M}_k\}$ that play a crucial role in our analysis. In the sequel we will work intensively with the mgf of $\boldsymbol{M}_k$ given $\boldsymbol{M}_{k-1}$ and $X_{k-1} = i$ (with $k \in \mathbb{N}$ and $i \in E$): for $\boldsymbol{s} \in \mathbb{R}^L$,

$$\xi_i(\boldsymbol{s} \,|\, \boldsymbol{m}) := \mathbb{E}[e^{\langle \boldsymbol{s}, \boldsymbol{M}_k \rangle} |\, \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i] = \mathbb{E}[e^{\langle \boldsymbol{s}, \boldsymbol{M}_1 \rangle} |\, \boldsymbol{M}_0 = \boldsymbol{m}, X_0 = i],$$

with the corresponding cgf $\zeta_i(\boldsymbol{s} \,|\, \boldsymbol{m}) := \log \xi_i(\boldsymbol{s} \,|\, \boldsymbol{m})$. Furthermore, we define for all observation pairs $\boldsymbol{m}, \boldsymbol{m}' \in \mathbb{N}_0^L$, $k \in \mathbb{N}$ and $i \in E$ the one-step transition probabilities

$$\begin{aligned} t_i(\boldsymbol{m}' \,|\, \boldsymbol{m}) &= \mathbb{P}(\boldsymbol{M}_k = \boldsymbol{m}' \,|\, \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i) \\ &= \mathbb{P}(\boldsymbol{M}_1 = \boldsymbol{m}' \,|\, \boldsymbol{M}_0 = \boldsymbol{m}, X_0 = i), \end{aligned}$$

and the diagonal matrix

$$T(\boldsymbol{m}' \,|\, \boldsymbol{m}) = \operatorname{diag}\big\{t_1(\boldsymbol{m}' \,|\, \boldsymbol{m}), \ldots, t_d(\boldsymbol{m}' \,|\, \boldsymbol{m})\big\}. \tag{5.3}$$

Note that $\xi_i(\boldsymbol{s} \,|\, \boldsymbol{m})$ and $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ do not depend on k due to time-homogeneity.

5.2.2 Parameter estimation

The objective of this chapter is to estimate the model parameters from observations of the population process. We now specify these unknown parameters and the available data.

Throughout we assume that the network population process $\{\boldsymbol{M}_k\}$ can be observed at time points $k = 0, 1, \ldots, n$ for some $n \in \mathbb{N}$. We denote the corresponding observations by $\boldsymbol{m}_0, \boldsymbol{m}_1, \ldots, \boldsymbol{m}_n$, so that the set $\{\boldsymbol{m}_0, \ldots, \boldsymbol{m}_n\} \in \mathbb{N}_0^{L \times (n+1)}$ comprises the available data.

Let

$$\theta = \Big(\alpha_i, p_{ij}, \lambda_i^{(\ell)}, r_i^{(\ell,\ell')} : i, j \in \{1, \ldots, d\},\, \ell \in \{1, \ldots, L\},\, \ell' \in \{0, \ldots, L\}\Big)^\top$$

be the unknown parameter vector corresponding to the model. Our goal is to estimate θ given the observation $\boldsymbol{m}_0, \ldots, \boldsymbol{m}_n$. The resulting estimate will be denoted by

$$\hat{\theta} = \Big(\hat{\alpha}_i, \hat{p}_{ij}, \hat{\lambda}_i^{(\ell)}, \hat{r}_i^{(\ell,\ell')} : i, j \in \{1, \ldots, d\},\, \ell \in \{1, \ldots, L\},\, \ell' \in \{0, \ldots, L\}\Big)^\top.$$

We estimate θ by maximum likelihood, which requires the evaluation of the likelihood function. We make the common assumption that $\mathbb{P}(\boldsymbol{M}_0 = \boldsymbol{m}_0) = 1$. By taking into account all possible paths of the background process $\{X_k\}$ (at times $k = 0, \ldots, n-1$), and using (5.3), the likelihood function can then be written as

$$\begin{aligned}
&\mathcal{L}(\theta \,|\, \boldsymbol{m}_0, \ldots, \boldsymbol{m}_n) = \mathbb{P}_\theta(\boldsymbol{M}_0 = \boldsymbol{m}_0, \ldots, \boldsymbol{M}_n = \boldsymbol{m}_n) \\
&= \sum_{x_0,\ldots,x_{n-1} \in E} \mathbb{P}_\theta(\boldsymbol{M}_0 = \boldsymbol{m}_0, X_0 = x_0, \ldots, \boldsymbol{M}_{n-1} = \boldsymbol{m}_{n-1}, X_{n-1} = x_{n-1}, \boldsymbol{M}_n = \boldsymbol{m}_n) \\
&= \boldsymbol{\alpha}^\top T(\boldsymbol{m}_1 \,|\, \boldsymbol{m}_0)\, P\, T(\boldsymbol{m}_2 \,|\, \boldsymbol{m}_1)\, P \cdots P\, T(\boldsymbol{m}_n \,|\, \boldsymbol{m}_{n-1})\, \mathbf{1},
\end{aligned} \tag{5.4}$$

where $\mathbf{1} = (1, \ldots, 1)^\top$. We conclude that, in order to compute the likelihood $\mathcal{L}(\theta \,|\, \boldsymbol{m}_0, \ldots, \boldsymbol{m}_n)$, it is a prerequisite to be able to evaluate, for any pair of vectors $\boldsymbol{m}'$ and $\boldsymbol{m}$ and for any $i \in E$, the probability $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$.

5.3 Small networks: explicit approach

In this section we present a few examples of 'small' networks in which the one-step probabilities $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ can be computed explicitly. We first consider the special case of a single-node model with Poisson arrivals, also known as a Markov-modulated infinite-server queue, and then treat a specific two-node tandem network. For larger networks, transitions from $\boldsymbol{m}$ to $\boldsymbol{m}'$ could correspond to a large number of potential scenarios (in terms of the numbers of individuals arriving, being routed to another node, and departing), making explicit evaluation prohibitive.

5.3.1 Single-node model

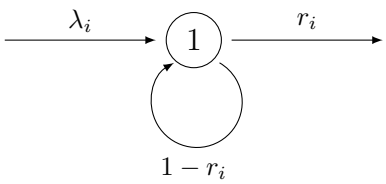

Figure 5.1: Schematic representation of the single-node model

Consider a model with a single node at which individuals arrive according to the arrival process $\{A_k\}_{k \in \mathbb{N}}$, which is now a univariate random variable. More precisely, $A_k \in \mathbb{N}_0$ is the number of arrivals in the k-th timeslot. Let, as before, $\{X_k\}_{k \in \mathbb{N}_0}$ be a Markovian background process with d states. We assume that for each state $i \in E$, A_k given $X_{k-1} = i$ has a Poisson distribution with parameter $\lambda_i \geqslant 0$, and individuals can either leave the

node with probability $r_i \in [0,1]$, or stay at the node with probability $1 - r_i$ (see Figure 5.1). Let the process $\{D_k\}_{k\in\mathbb{N}}$ count the number of individuals that leave the node per timeslot, whereas $\{M_k\}_{k\in\mathbb{N}_0}$ keeps track of the population size at the node. The idea is to compute $t_i(m' \,|\, m)$, by conditioning on the number of departing individuals at time $k = 1$. It follows that

$$\begin{aligned} t_i(m' \,|\, m) &= \sum_{\check{m}=0}^{m} \mathbb{P}(M_k = m' \,|\, D_k = \check{m}, M_{k-1} = m, X_{k-1} = i) \\ &\quad \cdot \mathbb{P}(D_k = \check{m} \,|\, M_{k-1} = m, X_{k-1} = i) \\ &= \sum_{\check{m}=\max\{0,m-m'\}}^{m} \frac{(\lambda_i)^{m'-(m-\check{m})}}{(m'-(m-\check{m}))!} e^{-\lambda_i} \binom{m}{\check{m}} (r_i)^{\check{m}} (1-r_i)^{m-\check{m}}. \end{aligned}$$

5.3.2 Tandem network

We now consider a tandem model with two nodes, in which individuals arrive at the first node, then either jump to the second node or stay at the first node, and from the second node either leave the system or stay at the second node. We again have a Markovian background process $\{X_k\}_{k\in\mathbb{N}_0}$ modulating the parameters in the model. We assume that individuals arrive at the first node according to the arrival process $\{A_k\}_{k\in\mathbb{N}}$, where A_k given $X_{k-1} = i$ is Poisson distributed with parameter $\lambda_i \geqslant 0$. Recall that $\boldsymbol{D}_k^{(1)}(2)$ represents the number of individuals jumping from the first to the second node. Given the state of the background process being $i \in E$, each individual makes this jump with probability $r_i^{(1,2)} \in [0,1]$, or stays at the first node with probability $1 - r_i^{(1,2)}$. From the second node, individuals leave the network with probability $r_i^{(2,0)} \in [0,1]$, or stay at the node with probability $1 - r_i^{(2,0)}$ (see Figure 5.2).

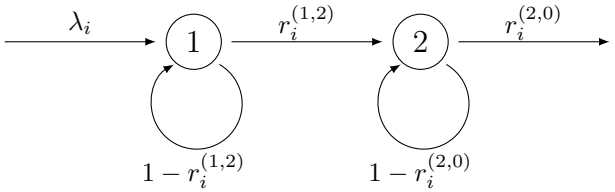

Figure 5.2: Schematic representation of the tandem network

We can compute $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ for this model, by conditioning on the number of individuals that jump from the first node to the second node. After some elementary algebra

we find

$$\begin{aligned} t_i(\boldsymbol{m}' \,|\, \boldsymbol{m}) &= \sum_{\check{m}=0}^{\boldsymbol{m}(1)} \mathbb{P}(\boldsymbol{M}_k = \boldsymbol{m}' \,|\, \boldsymbol{D}_k^{(1)}(2) = \check{m}, \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i) \\ &\quad \cdot \mathbb{P}(\boldsymbol{D}_k^{(1)}(2) = \check{m} \,|\, \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i) \\ &= \sum_{\check{m}=\check{m}_{\text{low}}}^{\check{m}_{\text{up}}} \frac{(\lambda_i)^a}{a!} e^{-\lambda_i} \binom{\boldsymbol{m}(2)}{b} (r_i^{(2,0)})^b (1 - r_i^{(2,0)})^{\boldsymbol{m}(2)-\check{m}} \\ &\quad \cdot \binom{\boldsymbol{m}(1)}{\check{m}} (r_i^{(1,2)})^{\check{m}} (1 - r_i^{(1,2)})^{\boldsymbol{m}'(1)-\check{m}}. \end{aligned} \tag{5.5}$$

Here $\check{m}_{\text{low}} := \max\{0, \boldsymbol{m}(1) - \boldsymbol{m}'(1), \boldsymbol{m}'(2) - \boldsymbol{m}(2)\}$, $\check{m}_{\text{up}} := \min\{\boldsymbol{m}(1), \boldsymbol{m}'(2)\}$ are the lower and upper bounds of the sum, respectively. In addition, $a := \boldsymbol{m}'(1) - \boldsymbol{m}(1) + \check{m}$ denotes the number of arrivals to the first node, and $b := \boldsymbol{m}(2) - \boldsymbol{m}'(2) + \check{m}$ the number of departures from the second node.

In the above two examples we observe that one can develop explicit expressions for $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$, but already in the example of the two-node tandem the expression becomes quite involved. When trying to extend our expressions to tandems with more nodes, or even to more general networks, the expressions will become increasingly complex as the dimension of the underlying network grows. As pointed out in e.g. [26], the computation effectively requires a complete enumeration over all possible configurations, which makes this explicit approach infeasible for larger networks. A solution to this problem for such networks is to, instead of pursuing exact calculation of $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$, resort to its *saddlepoint approximation*. We detail this procedure in the next section.

5.4 General networks: saddlepoint approximation

The main objective of this section is to set up an accurate and computationally efficient approximation for the probabilities $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$. As pointed out in Section 5.3, for multi-node models it is typically infeasible to evaluate $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ explicitly, which motivates the need for such approximative techniques. We rely on the saddlepoint approach [22, 60], which approximates a random variable's probability mass function through its mgf. In Section 5.4.1 we point out in detail how this technique works. A complication is that the saddlepoint machinery does not work for states at the boundary of the state space of $\{M_k\}$. For such points an alternative computation scheme is proposed in Section 5.4.2, which is a combination of the saddlepoint approximation with exact computations. Examples that assess the procedure's numerical performance are provided in Section 5.4.3.

5.4.1 Interior states: saddlepoint approach

Aiming at applying the saddlepoint approach to approximate $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$, we need to be able to evaluate the mgf $\xi_i(\boldsymbol{s} \,|\, \boldsymbol{m})$, where we recall the notation

$$\xi_i(\boldsymbol{s} \,|\, \boldsymbol{m}) = \mathbb{E}[e^{\langle \boldsymbol{s}, \boldsymbol{M}_k \rangle} | \, \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i].$$

The corresponding cgf is denoted by $\zeta_i(\boldsymbol{s} \,|\, \boldsymbol{m}) := \log \xi_i(\boldsymbol{s} \,|\, \boldsymbol{m})$. In order to evaluate $\xi_i(\boldsymbol{s} \,|\, \boldsymbol{m})$, observe that the ℓ-th component of $\boldsymbol{M}_k$ is equal to

- the number $\boldsymbol{M}_{k-1}(\ell)$ that was present at node ℓ at time $k-1$,
- decreased by the number of individuals that leave node ℓ at time k (either by jumping to another node or by leaving the network),
- increased by external arrivals at node ℓ at time k, and
- increased by the number of individuals that were at node $\check{\ell}$ at time $k-1$ and jump to node ℓ at time k, over all $\check{\ell} \in \{1, \ldots, L\}$.

Recall that $\boldsymbol{D}_k^{(\ell)}(L+1)$ represents the number of individuals that leave the network from node ℓ at time k. Summarizing the above, the following identity links $\boldsymbol{M}_k$ and $\boldsymbol{M}_{k-1}$:

$$\boldsymbol{M}_k(\ell) = \boldsymbol{M}_{k-1}(\ell) - \sum_{\check{\ell}=1}^{L+1} \boldsymbol{D}_k^{(\ell)}(\check{\ell}) + \boldsymbol{A}_k(\ell) + \sum_{\check{\ell}=1}^{L} \boldsymbol{D}_k^{(\check{\ell})}(\ell). \tag{5.6}$$

For ease of notation, both sums in (5.6) contain the variable $\boldsymbol{D}_k^{(\ell)}(\ell)$ corresponding to $\check{\ell} = \ell$, counting the number of individuals that stay at node ℓ. Recall that conditionally on $\boldsymbol{M}_{k-1} = \boldsymbol{m}$ and $X_{k-1} = i$, the vectors $\boldsymbol{D}_k^{(\ell)}$ are independent, and that for a given ℓ the entries of $\boldsymbol{D}_k^{(\ell)}$ have a multinomial distribution. Due to these properties and using (5.6), we find

$$\begin{aligned}
\xi_i(\boldsymbol{s} \,|\, \boldsymbol{m}) &= e^{\langle \boldsymbol{s}, \boldsymbol{m} \rangle} \, \phi_i(\boldsymbol{s}) \\
&\quad \cdot \mathbb{E}\Big[\exp\Big(\sum_{\ell=1}^{L} \boldsymbol{s}(\ell) \Big(\sum_{\check{\ell}=1}^{L} \boldsymbol{D}_k^{(\check{\ell})}(\ell) \, - \, \sum_{\check{\ell}=1}^{L+1} \boldsymbol{D}_k^{(\ell)}(\check{\ell}) \Big) \Big) \Big| \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i \Big] \\
&\qquad = e^{\langle \boldsymbol{s}, \boldsymbol{m} \rangle} \, \phi_i(\boldsymbol{s}) \\
&\quad \cdot \prod_{\check{\ell}=1}^{L} \mathbb{E}\Big[\exp\Big(\sum_{\ell=1}^{L} \boldsymbol{s}(\ell) \boldsymbol{D}_k^{(\check{\ell})}(\ell) \, - \, \sum_{\ell=1}^{L+1} \boldsymbol{s}(\check{\ell}) \boldsymbol{D}_k^{(\check{\ell})}(\ell) \Big) \Big| \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i \Big].
\end{aligned} \tag{5.7}$$

To obtain (5.7), we have used a change of summation in the first term of the exponent, a change of variables in the second term of the exponent, and the fact that the $\boldsymbol{D}_k^{(\ell)}(\check{\ell})$

are independent in ℓ. Continuing from (5.7), reordering the terms in the exponent, we conclude that we have

$$\xi_i(\boldsymbol{s}\,|\,\boldsymbol{m}) = e^{\langle \boldsymbol{s},\boldsymbol{m}\rangle}\,\phi_i(\boldsymbol{s}) \cdot \prod_{\check{\ell}=1}^{L} \mathbb{E}\left[\exp\Big(\sum_{\ell=1}^{L}(\boldsymbol{s}(\ell)-\boldsymbol{s}(\check{\ell}))\boldsymbol{D}_k^{(\check{\ell})}(\ell) \;-\; \boldsymbol{s}(\check{\ell})\boldsymbol{D}_k^{(\check{\ell})}(L+1)\Big)\Big|\boldsymbol{M}_{k-1}=\boldsymbol{m}, X_{k-1}=i\right].$$

Finally using the multinomial property, we arrive at the following result.

Lemma 1. For $\boldsymbol{s}\in\mathbb{R}^L$ and $\boldsymbol{m}\in\mathbb{N}_0^L$, and for any $i\in E$,

$$\begin{aligned}\xi_i(\boldsymbol{s}\,|\,\boldsymbol{m}) &= e^{\langle \boldsymbol{s},\boldsymbol{m}\rangle}\,\phi_i(\boldsymbol{s}) \prod_{\check{\ell}=1}^{L}\left(\sum_{\ell=1}^{L} r_i^{(\check{\ell},\ell)} e^{\boldsymbol{s}(\ell)-\boldsymbol{s}(\check{\ell})} + r_i^{(\check{\ell},0)} e^{-\boldsymbol{s}(\check{\ell})}\right)^{\boldsymbol{m}(\check{\ell})} \\ &= \phi_i(\boldsymbol{s}) \prod_{\check{\ell}=1}^{L}\left(\sum_{\ell=1}^{L} r_i^{(\check{\ell},\ell)} e^{\boldsymbol{s}(\ell)} + r_i^{(\check{\ell},0)}\right)^{\boldsymbol{m}(\check{\ell})}. \end{aligned} \tag{5.8}$$

Having the expression for $\xi_i(\boldsymbol{s}\,|\,\boldsymbol{m})$ at our disposal, we now point out how this can be used in the saddlepoint-based approximation of $t_i(\boldsymbol{m}'\,|\,\boldsymbol{m})$. To this end, we first note that by taking logarithms on both sides of Equation (5.8), we obtain

$$\zeta_i(\boldsymbol{s}\,|\,\boldsymbol{m}) = \psi_i(\boldsymbol{s}) + \sum_{\check{\ell}=1}^{L} \boldsymbol{m}(\check{\ell})\,\log\left(\sum_{\ell=1}^{L} r_i^{(\check{\ell},\ell)} e^{\boldsymbol{s}(\ell)} + r_i^{(\check{\ell},0)}\right). \tag{5.9}$$

It is known that any (joint) cgf is a convex function, which implies that $\zeta_i(\boldsymbol{s}\,|\,\boldsymbol{m})$ is convex (in $\boldsymbol{s}$). Define for $\boldsymbol{v},\boldsymbol{m}\in\mathbb{N}_0^L$, the corresponding multivariate *Legendre-Fenchel transforms* by

$$I_i(\boldsymbol{v}\,|\,\boldsymbol{m}) := \sup_{\boldsymbol{s}} I_i(\boldsymbol{v},\boldsymbol{s}\,|\,\boldsymbol{m}),$$

where $I_i(\boldsymbol{v},\boldsymbol{s}\,|\,\boldsymbol{m}) := \langle \boldsymbol{s},\boldsymbol{v}\rangle - \zeta_i(\boldsymbol{s}\,|\,\boldsymbol{m})$.

Let $S_i(\boldsymbol{m})\subseteq\mathbb{N}_0^L$ the set of states that can be reached from $\boldsymbol{m}$ in one time step when the background state is i. More concretely,

$$S_i(\boldsymbol{m}) = \left\{\boldsymbol{m}' : \boldsymbol{m}'(\ell) = \sum_{\ell'=1}^{L} k_{\ell',\ell} + k_\ell, \quad \big((k_{\ell',\ell})_{\ell',\ell=1}^{L},(k_\ell)_{\ell=1}^{L}\big)\in K_i(\boldsymbol{m})\right\},$$

where $K_i(\boldsymbol{m})$ is the subset of $\mathbb{N}_0^{L^2\times L}$ consisting of $\big((k_{\ell',\ell})_{\ell',\ell=1}^{L},(k_\ell)_{\ell=1}^{L}\big)$ such that

(A) For all $\ell=1,\ldots,L$, $\sum_{\ell'=1}^{L} k_{\ell,\ell'} \leqslant \boldsymbol{m}(\ell)$ (i.e., the sum of individuals leaving node ℓ cannot be more than $\boldsymbol{m}(\ell)$);

(B) For all $\ell=1,\ldots,L$, $\sum_{\ell'=1}^{L} k_{\ell,\ell'} = \boldsymbol{m}(\ell)$ if $r_i^{(\ell,0)}=0$ (i.e., the sum of individuals

jumping from node ℓ to the other nodes must be exactly $\boldsymbol{m}(\ell)$ if $r_i^{(\ell,0)} = 0$);

(C) For all $\ell = 1, \ldots, L$, $k_{\ell',\ell} = 0$ if $r_i^{(\ell',\ell)} = 0$, and $k_\ell = 0$ if $\lambda_i^{(\ell)} = 0$ (i.e., jumps and arrivals cannot occur if the corresponding parameter equals zero).

We denote by $S_i(\boldsymbol{m})^\circ$ the 'interior' of $S_i(\boldsymbol{m})$, to be understood as $S_i(\boldsymbol{m})$ minus its boundaries.

For any $\boldsymbol{v} \in S_i(\boldsymbol{m})^\circ$ there is a unique optimizing vector $\boldsymbol{s}_{\boldsymbol{v}}^*$ for which $I_i(\boldsymbol{v}, \boldsymbol{s}_{\boldsymbol{v}}^* \,|\, \boldsymbol{m}) = I_i(\boldsymbol{v} \,|\, \boldsymbol{m})$, which is called the saddlepoint; see [17, Chapter 1] for more details. By the definition of $I_i(\boldsymbol{v}, \boldsymbol{s} \,|\, \boldsymbol{m})$, this saddlepoint is the unique solution of the system of L first-order conditions

$$\boldsymbol{v}(\ell') = \frac{\partial \psi_i(\boldsymbol{s})}{\partial \boldsymbol{s}(\ell')} + \sum_{\check{\ell}=1}^{L} \boldsymbol{m}(\check{\ell})\, r_i^{(\check{\ell},\ell')} e^{\boldsymbol{s}(\ell')} \Big/ \Big(\sum_{\ell=1}^{L} r_i^{(\check{\ell},\ell)} e^{\boldsymbol{s}(\ell)} + r_i^{(\check{\ell},0)} \Big), \tag{5.10}$$

where the right hand side of (5.10) is the ℓ'-th entry of the gradient of the cgf, that is the vector of first partial derivatives with respect to the entries of $\boldsymbol{s}$. Let $\Sigma_i(\boldsymbol{v} \,|\, \boldsymbol{m})$ be the $L \times L$ Hessian matrix of the cgf evaluated at the saddlepoint with (ℓ', ℓ'')-th entry given by

$$\Sigma_i^{(\ell',\ell'')}(\boldsymbol{v} \,|\, \boldsymbol{m}) = \left. \frac{\partial^2 \zeta_i(\boldsymbol{s} \,|\, \boldsymbol{m})}{\partial \boldsymbol{s}(\ell')\, \partial \boldsymbol{s}(\ell'')} \right|_{\boldsymbol{s}=\boldsymbol{s}_{\boldsymbol{v}}^*}.$$

Note that by taking another partial derivative of the right hand side of (5.10), we find for $\ell' \neq \ell''$

$$\frac{\partial^2 \zeta_i(\boldsymbol{s} \,|\, \boldsymbol{m})}{\partial \boldsymbol{s}(\ell')\, \partial \boldsymbol{s}(\ell'')} = \frac{\partial^2 \psi_i(\boldsymbol{s})}{\partial \boldsymbol{s}(\ell')\, \partial \boldsymbol{s}(\ell'')} + \sum_{\check{\ell}=1}^{L} \boldsymbol{m}(\check{\ell}) \frac{-r_i^{(\check{\ell},\ell')} e^{\boldsymbol{s}(\ell')} r_i^{(\check{\ell},\ell'')} e^{\boldsymbol{s}(\ell'')}}{\Big(\sum_{\ell=1}^{L} r_i^{(\check{\ell},\ell)} e^{\boldsymbol{s}(\ell)} + r_i^{(\check{\ell},0)} \Big)^2},$$

while for $\ell' = \ell''$

$$\frac{\partial^2 \zeta_i(\boldsymbol{s} \,|\, \boldsymbol{m})}{\partial \boldsymbol{s}(\ell')\, \partial \boldsymbol{s}(\ell'')} = \frac{\partial^2 \psi_i(\boldsymbol{s})}{\partial \boldsymbol{s}^2(\ell')} + \sum_{\check{\ell}=1}^{L} \boldsymbol{m}(\check{\ell}) \frac{r_i^{(\check{\ell},\ell')} e^{\boldsymbol{s}(\ell')} \big(\sum_{\ell \neq \ell'} r_i^{(\check{\ell},\ell)} e^{\boldsymbol{s}(\ell)} + r_i^{(\check{\ell},0)} \big)}{\Big(\sum_{\ell=1}^{L} r_i^{(\check{\ell},\ell)} e^{\boldsymbol{s}(\ell)} + r_i^{(\check{\ell},0)} \Big)^2}.$$

We can now present the saddlepoint approximation [17, 22]. In the statement below, $|D|$ denotes the determinant of the matrix D.

Approximation 5.1. For $\boldsymbol{m} \in \mathbb{N}_0^L$, $\boldsymbol{m}' \in S(\boldsymbol{m})^\circ$, and for any $i \in E$, the saddlepoint approximation of $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ is given by

$$t_i(\boldsymbol{m}' \,|\, \boldsymbol{m}) \approx (2\pi)^{-L/2} \, |\, \Sigma_i(\boldsymbol{m}' \,|\, \boldsymbol{m})|^{-1/2} \exp\big(- I_i(\boldsymbol{m}' \,|\, \boldsymbol{m}) \big). \tag{5.11}$$

Observe that the complexity of evaluating this approximation is relatively low. More specifically, to evaluate $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ the maximization of an L-dimensional concave function

needs to be performed and the determinant of a $(L \times L)$-matrix needs to be computed. To evaluate the full (diagonal) matrix $T(\boldsymbol{m}' \,|\, \boldsymbol{m})$, this has to be done d times. The computation of the likelihood $\mathcal{L}(\theta \,|\, \boldsymbol{m}_0, \ldots, \boldsymbol{m}_n)$ then takes $2n$ matrix multiplications, with matrices of size $d \times d$, where n of these multiplications can be done relatively efficiently as they involve a diagonal matrix.

Remark 1. In the model considered, individuals jump between nodes until they leave the network. Interestingly, a 'branching variant' of this model, in which there is the option of a single individual splitting into multiple individuals, can also be dealt with. This variant is also referred to as a multitype branching process with immigration in a random environment; see [39, 62] for an analysis of its limiting distribution. In this case, when an individual moves from ℓ to ℓ' (with the background process being in state i), the number of individuals that end up at ℓ' is not necessarily 1, but is distributed as a random variable $W_i^{(\ell,\ell')} \in \mathbb{N}_0$ with mgf $w_i^{(\ell,\ell')}(s)$ (assumed to exist). Then for $\boldsymbol{s} \in \mathbb{R}^L$ and $\boldsymbol{m} \in \mathbb{N}_0^L$, and for any $i \in E$, the mgf $\xi_i(\boldsymbol{s} \,|\, \boldsymbol{m})$ becomes

$$\xi_i(\boldsymbol{s} \,|\, \boldsymbol{m}) = \phi_i(\boldsymbol{s}) \prod_{\ell'=1}^{L} \left(\sum_{\ell=1}^{L} r_i^{(\ell',\ell)} w_i^{(\ell',\ell)}(\boldsymbol{s}(\ell)) + r_i^{(\ell',0)} \right)^{\boldsymbol{m}(\ell')}.$$

Observe that the resulting network is not necessarily stable; we do not further comment on the stability condition of this model. In another variant that can be dealt with, each individual that leaves ℓ can potentially cause arrivals at *all* nodes simultaneously, rather than at just one node.

5.4.2 States at the boundaries

Above we introduced an approximation for $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ with $\boldsymbol{m}' \in S_i(\boldsymbol{m})^\circ$, which leaves us with the question what should be done for the 'boundary points' $\boldsymbol{m}' \in S_i(\boldsymbol{m}) \setminus S_i(\boldsymbol{m})^\circ$. In the first place we recall (see [17, Chapter 1]) that for these points the saddlepoint approximation cannot be used, as a consequence of the fact that the optimizing $\boldsymbol{s}^*_{\boldsymbol{m}'}$ cannot be determined. To show how we remedy this, we first use the illustrative examples of the single-node model and the tandem network featured in Section 5.3. As we will observe, in these cases the transition probabilities can be found explicitly for the boundary states. Later in this subsection we will set up a general (exact) procedure to compute the transition probabilities for boundary states.

- For the single-node model, $S_i(m) = \mathbb{N}_0$, and thus $S_i(m) \setminus S_i(m)^\circ = \{0\}$. Now consider $m' = 0$. For this boundary point an easy explicit expression for $t_i(m' \,|\, m)$ can be given. We have the explicit expression

  $$t_i(0 \,|\, m) = e^{-\lambda_i} (r_i)^m,$$

  since there should be no new arrivals, and all individuals that were present at the node have to leave.

- We continue by considering the tandem network. A first observation is that for this network there are multiple boundary points to take into account. There are no external arrivals at the second node, because this node is only fed by individuals moving from the first to the second node. As a consequence, we have

$$S_i(\boldsymbol{m}) = \left\{\boldsymbol{m}' \in \mathbb{N}_0^2 : \max\{0, \boldsymbol{m}(1) - \boldsymbol{m}'(1)\} \leqslant \boldsymbol{m}'(2) \leqslant \boldsymbol{m}(1) + \boldsymbol{m}(2)\right\}.$$

  To verify this, note that the maximum number of individuals at the second node at time k cannot be larger than the total network population at time $k-1$, and the minimum number of individuals cannot be smaller than the minimum inflow from node 1.

  Now consider a boundary point $\boldsymbol{m}'$ in $S(\boldsymbol{m}) \setminus S(\boldsymbol{m})^\circ$. The claim is that again for all these boundary points an easy explicit expression for $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ can be given. It is for example readily checked that, in self-evident notation,

$$\begin{aligned} &t_i\big((\boldsymbol{m}'(1), \boldsymbol{m}(1) + \boldsymbol{m}(2))^\top \,|\, \boldsymbol{m}\big) \\ &\quad = \mathbb{P}\big(\mathrm{Pois}(\lambda_i) = \boldsymbol{m}'(1)\big)\, \big(r_i^{(1,2)}\big)^{\boldsymbol{m}(1)} \big(1 - r_i^{(2,0)}\big)^{\boldsymbol{m}(2)}. \end{aligned}$$

  Notice that this probability corresponds to a scenario in which all individuals present at node 1 have to move to node 2, and all those present at node 2 have to stay. Importantly, in this case the complicated combinatorial expression (5.5) reduces to a considerably easier expression, essentially due to the fact that at boundary points the transition corresponds to a very specific scenario.

With the above examples in mind, let us go back to the general network setting with L nodes. For ease we restrict ourselves to the situation where $\lambda_i^{(\ell)} > 0$ and $r_i^{(\ell,0)} > 0$ for all $i \in \{1, \ldots, d\}$ and all $\ell \in \{1, \ldots, L\}$. This means that at each node external arrivals and departures are possible for all states of the background process. The immediate consequence is that

$$S_i(\boldsymbol{m}) \setminus S_i(\boldsymbol{m})^\circ = \{\boldsymbol{m}' : \exists \ell \in \{1, \ldots, L\} : \boldsymbol{m}'(\ell) = 0\}.$$

The situation in which some of the $\lambda_i^{(\ell)}$ and $r_i^{(\ell,0)}$ are 0 requires a bit more administration, but can be handled similarly (as in the above tandem example). Now fix an $\boldsymbol{m}' \in S_i(\boldsymbol{m}) \setminus S_i(\boldsymbol{m})^\circ$, a boundary point. We define by $N(\boldsymbol{m}')$ all nodes of the new configuration $\boldsymbol{m}'$ that contain zero individuals, i.e., $N(\boldsymbol{m}') := \{\ell \in \{1, \ldots, L\} : \boldsymbol{m}'(\ell) = 0\}$. Let $E(\boldsymbol{m}')$ be the corresponding event defined as $E(\boldsymbol{m}') := \{\forall \ell \in N(\boldsymbol{m}'), \boldsymbol{M}_k(\ell) = 0\}$. Then,

because $\{\boldsymbol{M}_k = \boldsymbol{m}'\} \subseteq E(\boldsymbol{m}')$, and using elementary rules for conditional probabilities,

$$\begin{aligned} t_i(\boldsymbol{m}' \,|\, \boldsymbol{m}) &= \mathbb{P}(\boldsymbol{M}_k = \boldsymbol{m}' \,|\, \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i) \\ &= \mathbb{P}(\boldsymbol{M}_k = \boldsymbol{m}' \,|\, E(\boldsymbol{m}'), \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i) \\ &\quad \cdot \mathbb{P}(E(\boldsymbol{m}') \,|\, \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i). \end{aligned} \tag{5.12}$$

For the boundary points $\boldsymbol{m}'$, $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ can be (approximately) evaluated by evaluating the two factors in (5.12) separately. As we will see, the second factor can be computed exactly, whereas for the first one we can set up a saddlepoint approximation in the way demonstrated in Section 5.4.1.

To evaluate the second factor in (5.12), we observe that (i) at time 1, no arrivals are allowed in the nodes of $N(\boldsymbol{m}')$, and (ii) individuals present at the nodes in $\{1, \ldots, L\}$ at time 0 should either leave the network or move to a node in the complement of $N(\boldsymbol{m}')$. More specifically, they cannot move to, or stay in, a node in $N(\boldsymbol{m}')$. As a consequence, we have the exact expression

$$\mathbb{P}(E(\boldsymbol{m}') \,|\, \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i) = \prod_{\ell \in N(\boldsymbol{m}')} e^{-\lambda_i^{(\ell)}} \cdot \prod_{\ell=1}^{L} \Bigg(\sum_{\ell' \notin N(\boldsymbol{m}')} r_i^{(\ell,\ell')} + r_i^{(\ell,0)} \Bigg)^{m(\ell)}. \tag{5.13}$$

We now concentrate on the first factor in (5.12), which can be computed using a saddlepoint approximation. To this end, we first observe that the occurrence of the event $E(\boldsymbol{m}')$ (i.e., $\boldsymbol{M}_k(\ell) = 0$ for all $\ell \in N(\boldsymbol{m}')$) changes the distribution of the random vectors $\boldsymbol{A}_k$ and $\boldsymbol{D}_k$; in the sequel we denote the random vectors under this condition by $\tilde{\boldsymbol{A}}_k$ and $\tilde{\boldsymbol{D}}_k$. To describe the distribution of $\tilde{\boldsymbol{A}}_k$ and $\tilde{\boldsymbol{D}}_k$, we use the following 'renormalized' probabilities, for $\ell'' \notin N(\boldsymbol{m}')$:

$$\tilde{r}_i^{(\ell',\ell'')} = \frac{r_i^{(\ell',\ell'')}}{\displaystyle\sum_{\ell'' \notin N(\boldsymbol{m}')} r_i^{(\ell',\ell'')} + r_i^{(\ell',0)}}, \quad \tilde{r}_i^{(\ell',0)} = \frac{r_i^{(\ell',0)}}{\displaystyle\sum_{\ell'' \notin N(\boldsymbol{m}')} r_i^{(\ell',\ell'')} + r_i^{(\ell',0)}}.$$

We then make the following observations.

- Since we have independent Markov-modulated Poisson arrivals at each of the nodes, the components of $\tilde{\boldsymbol{A}}_k$ are independent, with $\tilde{\boldsymbol{A}}_k(\ell)$ having a Poisson distribution with parameter $\lambda_i^{(\ell)}$ for all $\ell \notin N(\boldsymbol{m}')$, whereas $\tilde{\boldsymbol{A}}_k(\ell) \equiv 0$ for all $\ell \in N(\boldsymbol{m}')$. Recall that no arrivals are allowed in the nodes of $N(\boldsymbol{m}')$ due to the condition imposed.

- The random vectors $\tilde{\boldsymbol{D}}_k^{(\ell')}$, for $\ell' = 1, \ldots, L$, are independent. More specifically $\tilde{\boldsymbol{D}}_k^{(\ell')}$ has a multinomial distribution that attains values in the complement of $N(\boldsymbol{m}')$ or $\{L+1\}$, where the latter option corresponds to leaving the network, with its

parameters being given by $\boldsymbol{m}(\ell')$ and the probabilities

$$\Big((\tilde{r}_i^{(\ell',\ell'')})_{\ell''\notin N(\boldsymbol{m}')},\ \tilde{r}_i^{(\ell',0)}\Big).$$

Recall that individuals present at any of the nodes should either leave the network or move to (or stay at) a node in the complement of $N(\boldsymbol{m}')$.

Similar to (5.6), conditionally on $E(\boldsymbol{m}')$, we thus have the representation

$$\boldsymbol{M}_k(\ell) = \boldsymbol{M}_{k-1}(\ell) + \tilde{\boldsymbol{A}}_k(\ell) + \sum_{\check{\ell}=1}^{L} \tilde{\boldsymbol{D}}_k^{(\check{\ell})}(\ell) - \sum_{\check{\ell}=1}^{L+1} \tilde{\boldsymbol{D}}_k^{(\ell)}(\check{\ell}).$$

We can now proceed as in Section 5.4.1, to obtain the mgf of $\boldsymbol{M}_k$, conditionally on the event $\{E(\boldsymbol{m}'), \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i\}$. Using the above findings, we find that it equals, with $\boldsymbol{s}$ now being a vector with zeroes at the positions that correspond to the elements in $N(\boldsymbol{m}')$,

$$\begin{aligned}\mathbb{E}[e^{\langle \boldsymbol{s},\boldsymbol{M}_k\rangle} \,|\, E(\boldsymbol{m}'), \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i] \\ = \tilde{\phi}_i(\boldsymbol{s}) \prod_{\check{\ell}=1}^{L} \Big(\sum_{\ell\notin N(\boldsymbol{m}')} \tilde{r}_i^{(\check{\ell},\ell)} e^{\boldsymbol{s}(\ell)} + \tilde{r}_i^{(\check{\ell},0)} \Big)^{\boldsymbol{m}(\check{\ell})},\end{aligned}$$

where

$$\tilde{\phi}_i(\boldsymbol{s}) = \prod_{\ell\notin N(\boldsymbol{m}')} e^{\lambda_i^{(\ell)}(e^{\boldsymbol{s}(\ell)}-1)}.$$

Observe in particular the similarity with the result stated in Lemma 1. Using this mgf, we can use a saddlepoint technique to approximate $\mathbb{P}(\boldsymbol{M}_k = \boldsymbol{m}' \,|\, E(\boldsymbol{m}'), \boldsymbol{M}_{k-1} = \boldsymbol{m}, X_{k-1} = i)$ in (5.12) by following the same argument as in Section 5.4.1, evidently only including the non-zero elements of $\boldsymbol{m}'$. We observe that the dimension of this saddlepoint approximation is now $L - \#\{N(\boldsymbol{m}')\}$, which is smaller than L as a consequence of $\boldsymbol{m}' \in S_i(\boldsymbol{m}) \setminus S_i(\boldsymbol{m})^\circ$.

In summary, according to (5.12) the probability $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ can be factorized into two probabilities. The probability corresponding to the nodes included in $N(\boldsymbol{m}')$ can be computed explicitly according to (5.13), whereas the probability corresponding to the remaining nodes can be evaluated relying on the saddlepoint approximation of reduced dimension.

5.4.3 Numerical assessment of approximations

We can illustrate the accuracy of the saddlepoint approximation for the single-node model and the tandem network, by comparing the explicit approach from Section 5.3 with the saddlepoint approach from Section 5.4.

Example 5.1. *Single-node model.* Consider the example of the single-node model introduced in Section 5.3.1, where we computed $t_i(m' \,|\, m)$ explicitly. In this example $S_i(m) = \mathbb{N}$, assuming that $\lambda_i > 0$ and $r_i > 0$, so that $S_i(m) \setminus S_i(m)^\circ = \{0\}$. Using the saddlepoint approach we can, for $m' \in S_i(m)^\circ$, approximate $t_i(m' \,|\, m)$ using Approximation 1. Recall from (5.2) that for the Poisson arrivals at the node we have $\psi_i(s) = \lambda_i(e^s - 1)$. Using (5.9) with $r_i^{(1,1)} = 1 - r_i$ and $r_i^{(1,0)} = r_i$, we find the cgf

$$\zeta_i(s \,|\, m) = \lambda_i(e^s - 1) + m \log\big((1 - r_i)e^s + r_i\big).$$

It requires a few standard steps to verify that the saddlepoint $\boldsymbol{s}_{\boldsymbol{v}}^*$ can be found by solving

$$v = w(s) := \lambda_i e^s + m\frac{(1 - r_i)e^s}{(1 - r_i)e^s + r_i}; \tag{5.14}$$

observe that the right-hand side of (5.14) is a positive, increasing function in s, with $w(s) \to 0$ as $s \to -\infty$ and $w(s) \to \infty$ as $s \to \infty$. This means that for any $v > 0$, there is a unique solution s_v^*. More concretely, $e^{s_v^*}$ can be found in a standard manner by solving the quadratic equation

$$-\lambda_i(1 - r_i)e^{2s} + \big(v(1 - r_i) - \lambda_i r_i - m(1 - r_i)\big)e^s + vr_i = 0.$$

In our one-dimensional context we have

$$\Sigma_i(v \,|\, m) = \left.\frac{\partial^2 \zeta_i(s \,|\, m)}{\partial s^2}\right|_{s = s_v^*}.$$

Using that (5.14) holds when $s = s_v^*$, we thus find

$$\Sigma_i(v \,|\, m) = \frac{\lambda_i(1 - r_i)e^{2s_v^*} + vr_i}{(1 - r_i)e^{s_v^*} + r_i}$$

We have now collected all ingredients to evaluate the saddlepoint approximation (5.11). Concerning $m' \in S_i(m) \setminus S_i(m)^\circ = \{0\}$, we evidently have $t_i(0 \,|\, m) = e^{-\lambda_i}\,(r_i)^m$ as we saw before.

In Figure 5.3 we show the numerically obtained approximation in the single-node setting. It displays three examples which provide a good reflection of the accuracy typically achieved by the saddlepoint approach. In particular, they illustrate that the accuracy improves as the value of m increases, which is a known feature of saddlepoint approximations.

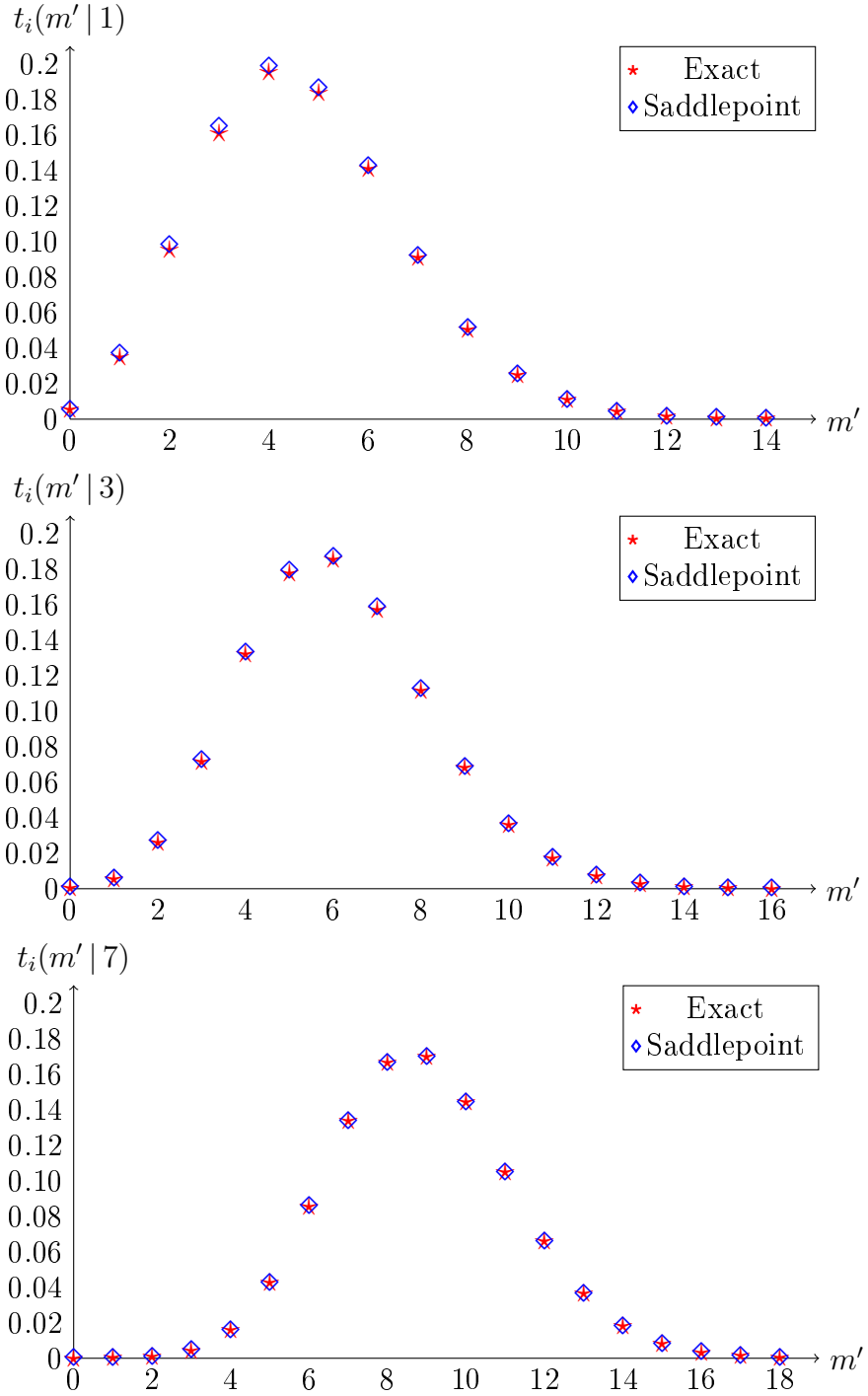

Figure 5.3: Saddlepoint approximation and exact computation of $t_i(m' \mid m)$ for the single-node model as a function of m', for increasing values of m; from the top to bottom panel, $m = 1, m = 3$ and $m = 7$. Parameter values: $i = 1, \lambda_1 = 4$ and $r_1 = 0.3$.

Example 5.2. *Tandem network.* To further assess the accuracy of the saddlepoint approximation, we consider the example of the tandem network with two nodes, as introduced in Section 5.3.2. We compute $t_i(\boldsymbol{m}' \,|\, \boldsymbol{m})$ explicitly, and compare it with its saddlepoint-based counterpart. We do this for $\boldsymbol{m}' \in S_i(\boldsymbol{m})^\circ$; note that we already discussed above how to deal with the boundary points $\boldsymbol{m}' \in S_i(\boldsymbol{m}) \setminus S_i(\boldsymbol{m})^\circ$. We can compute the cumulant generating function $\zeta_i(\boldsymbol{s} \,|\, \boldsymbol{m})$ from (5.9). From the fact that we have Poisson arrivals, we know that $\psi_i(\boldsymbol{s})$ follows from (5.2). The cgf equals

$$\begin{aligned}\zeta_i(\boldsymbol{s} \,|\, \boldsymbol{m}) =& \lambda_i(e^{\boldsymbol{s}(1)} - 1) + \boldsymbol{m}(1) \log\Big((1 - r_i^{(1,2)})e^{\boldsymbol{s}(1)} + r_i^{(1,2)} e^{\boldsymbol{s}(2)}\Big) \\ &+ \boldsymbol{m}(2) \log\Big((1 - r_i^{(2,0)})e^{\boldsymbol{s}(2)} + r_i^{(2,0)}\Big).\end{aligned}$$

Hence, for $v \in S_i(m)^\circ$ the saddlepoint $\boldsymbol{s}_{\boldsymbol{v}}^*$ can be found by solving the equations

$$\begin{aligned}\boldsymbol{v}(1) &= \lambda_i e^{\boldsymbol{s}(1)} + \boldsymbol{m}(1) \frac{(1 - r_i^{(1,2)})e^{\boldsymbol{s}(1)}}{(1 - r_i^{(1,2)})e^{\boldsymbol{s}(1)} + r_i^{(1,2)} e^{\boldsymbol{s}(2)}} \\ \boldsymbol{v}(2) &= \boldsymbol{m}(1) \frac{r_i^{(1,2)} e^{\boldsymbol{s}(2)}}{(1 - r_i^{(1,2)})e^{\boldsymbol{s}(1)} + r_i^{(1,2)} e^{\boldsymbol{s}(2)}} + \boldsymbol{m}(2) \frac{(1 - r_i^{(2,0)})e^{\boldsymbol{s}(2)}}{(1 - r_i^{(2,0)})e^{\boldsymbol{s}(2)} + r_i^{(2,0)}}.\end{aligned}$$

Having found the solution $\boldsymbol{s}_{\boldsymbol{v}}^*$, the approximation (5.11) is readily evaluated.

Numerical results for a few representative examples are presented in Figure 5.4. The upper panel in Figure 5.4 shows a cross section at the peak of the joint distribution of $\boldsymbol{m}'(1)$ and $\boldsymbol{m}'(2)$, the middle panel shows a cross section close to the peak, and the bottom panel shows a cross section further away from the peak. Our findings confirm the approach's high accuracy that we observed earlier.

5.5 Parameter estimation

In this section, we show how the saddlepoint approximation of the likelihood—developed in the previous section—can be used to estimate the model parameters, and we assess the accuracy of this estimation method by applying it to simulated data.

As argued before, we can use the saddlepoint approximation in (5.11) to approximate the probabilities $t_i(\boldsymbol{m}_k \,|\, \boldsymbol{m}_{k-1})$ for each pair of observations $(\boldsymbol{m}_{k-1}, \boldsymbol{m}_k)$ $(k = 1, \ldots, n)$ and each $i \in E$, so as to evaluate the likelihood (5.4). This likelihood is then to be maximized over the model parameters (in the appropriate parameter space) to find the parameter estimate $\hat{\theta}$. We do this numerically, relying on the built-in solver *fmincon* of MATLAB.

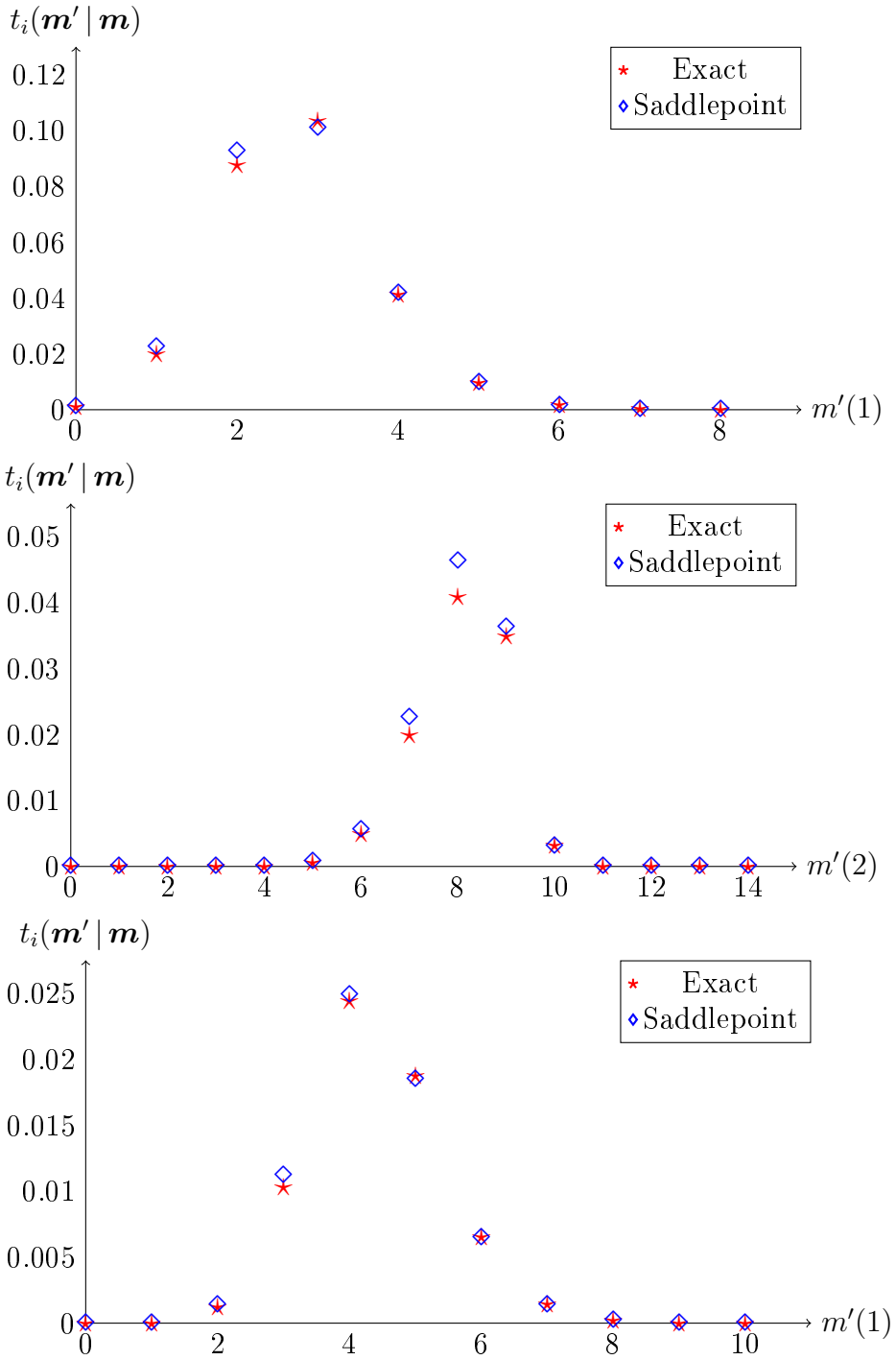

Figure 5.4: Saddlepoint approximation and exact computation of $t_i(\boldsymbol{m}' \mid \boldsymbol{m})$ for the tandem network as a function of m'. Parameter values: $i = 1, \lambda_1 = 0.5, r_1^{(1,2)} = 0.5$ and $r_1^{(2,0)} = 0.2$. Throughout we have fixed $\boldsymbol{m} = (5,5)^\top$. Upper panel: we vary $\boldsymbol{m}'(1)$, with $\boldsymbol{m}'(2) = 7$. Middle panel: we vary $\boldsymbol{m}'(2)$, with $\boldsymbol{m}'(1) = 1$. Bottom panel: we vary $\boldsymbol{m}'(1)$, with $\boldsymbol{m}'(2) = 4$.

The solver *fmincon* needs an initial value for θ. There are various ways to choose this value.

- In case the parameter space is finite, a naïve approach would be to sample the initial value uniformly on the parameter space.
- Another approach is to let the routing be uniform, in the sense that for any individual all next nodes are equally likely; for example in a fully connected graph (i.e., the situation that all $r_i^{(\ell,\ell')}$ are positive), we could set $r_i^{(\ell,1)} = r_i^{(\ell,2)} = \cdots = r_i^{(\ell,L)} = r_i^{(\ell,0)} = (L+1)^{-1}$ for all $i \in E$ and $\ell = 1, \ldots, L$. Likewise, the transition probabilities p_{ij} could be initialized with $1/d$.
- Alternatively, the initial θ can be determined using moment estimators, or, if available, additional information on the parameters can be used to set a suitable initial value.

In the remainder of this section, we specify the initial values that we used for each numerical experiment. As is commonly known, the maximum likelihood approach has the intrinsic issue that there can be local maxima. It is therefore strongly advised to follow the usual procedure to work with multiple initial values (and to record the one providing the highest likelihood).

Remark 2. Observe that for example in a model with $d = 2$, swapping the states in the parametrization results in an observationally equivalent model. In case of such identifiability issues, additional constraints need to be imposed on the parameters. In the single-node case of $d = 2$ with an environment-dependent arrival rate, such a constraint could for instance be $\lambda_2 \geq \lambda_1$.

Remark 3. We note that, by the structure of expression (5.4), the evaluation of the likelihood is linear in n and cubic in d. The complexity of the saddlepoint-based approximation is relatively low, due to the concavity of the functions $I_i(\boldsymbol{v}, \boldsymbol{s} \,|\, \boldsymbol{m})$.

To illustrate the broad applicability of the method, we perform numerical experiments for a set of intrinsically different networks. We specifically investigate the influence of the number of observations n on the estimates: throughout, we evaluate the estimators for $n = 100, n = 500, n = 1000$, and $n = 2000$. For each network and each value of n, we simulate 100 data sets, to each of which we apply the estimation method to obtain the parameter estimates. We present and discuss our findings in this section. We use the two examples from Sections 5.3 and 5.4, i.e., the single-node and the tandem, but we start with an experiment featuring a larger network with a different structure: a circle network.

Experiment 5.1. *Circle network.* We consider a network of five nodes in a circle. The individuals can move clockwise through the network from one node to the next. In this experiment we primarily concentrate on the effect of the network structure, and therefore we do not impose modulation (i.e., we consider the setting $d = 1$). In addition, we let the network be homogeneous, in the sense that the arrival processes, the probabilities of leaving the network, and the probabilities of being forwarded to the next node, respectively, are the same for any node. More concretely, we work with three parameters

$\lambda_1^{(1)} = \ldots = \lambda_1^{(5)} := \lambda$, $r_1^{(1,0)} = \ldots = r_1^{(5,0)} := r^0$, and $r_1^{(1,2)} = r_1^{(2,3)} = \ldots = r_1^{(5,1)} := r^1$. This means that *at each node* arrivals occur according to a Poisson process with rate λ, and any individual present at the node leaves with probability r^0, or jumps to the following node in the circle with probability r^1. Note that, as a result, individuals stay at a node with probability $1 - r^0 - r^1$; see Figure 5.5 for a pictorial illustration.

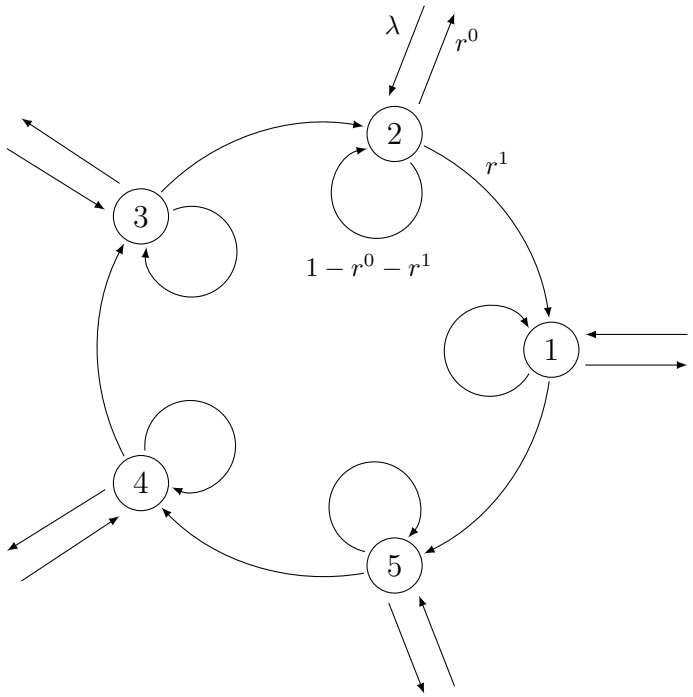

Figure 5.5: Schematic representation of the circle network of 5 nodes

Despite the fact that there is no modulation, direct evaluation of the likelihood is challenging. As pointed out earlier, the high complexity essentially lies in the fact that we observe the network population vector only, and not the arrival, routing and departure processes. An exact evaluation of the likelihood would require taking into account all paths of the arrival, routing and departure processes that match with the observed values of the network population vector, which for our five-node circle network would be infeasible. This motivates why we resort to evaluating the likelihood using the saddlepoint approximation.

In our experiments we use simulated data that are generated using the parameter values $\lambda = 1.5, r^1 = 0.3$, and $r^0 = 0.1$. The maximum likelihood estimation procedure using *fmincon* is initialized at $\lambda = 1, r^1 = \frac{1}{3}$ and $r^0 = \frac{1}{3}$. Experiments with other initial values lead to similar results. The numerical output is shown in Table 5.1 and Figures 5.6–5.8. Table 5.1 contains, for each sample size (rows) and parameter (columns), the mean value of the 100 estimates, together with the corresponding standard deviation between brackets. We see that the mean values in Table 5.1 lie close to the true parameter values, and that (as expected) the standard deviations decrease as n increases. This is visible in the histograms as well, displayed in Figures 5.6–5.8. Each figure shows, for a given value of n and one of the three parameters, the histogram of the 100 estimates. For each of the three parameters, we intentionally chose the same horizontal axis in all four pictures, so

as to provide insight into the speed at which the width of the peak decreases as n grows.

n	λ	r^1	r^0
100	1.4902 (0.4319)	0.3143 (0.0317)	0.1003 (0.0271)
500	1.5081 (0.1613)	0.3070 (0.0131)	0.1006 (0.0104)
1000	1.5197 (0.1034)	0.3059 (0.0086)	0.1013 (0.0062)
2000	1.5103 (0.0679)	0.3072 (0.0057)	0.1007 (0.0045)

Table 5.1: *Circle network*: mean of estimates of 100 data sets, with corresponding standard deviation between brackets. True parameter values: $\lambda = 1.5, r^1 = 0.3, r^0 = 0.1$.

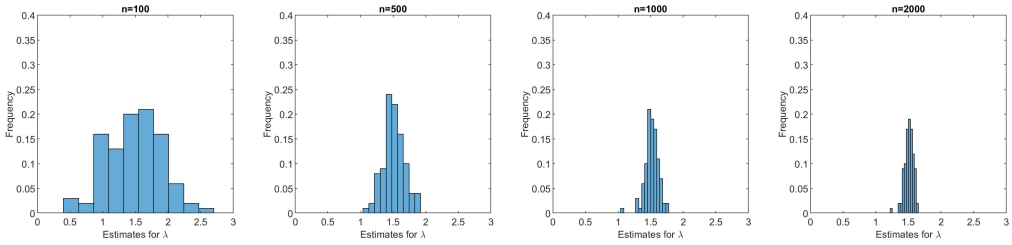

Figure 5.6: *Circle network*: histograms of the obtained estimates for λ, with n increasing from left to right.

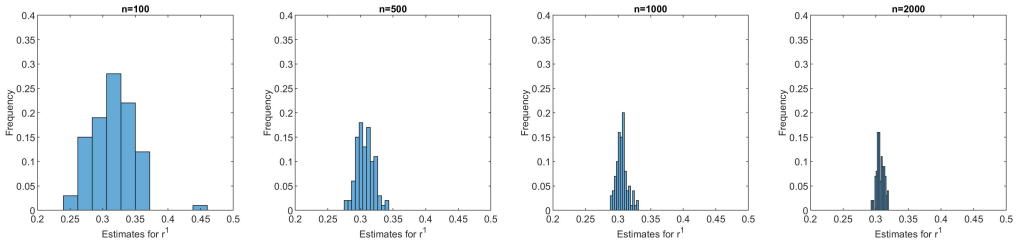

Figure 5.7: *Circle network*: histograms of the obtained estimates for r^1, with n increasing from left to right.

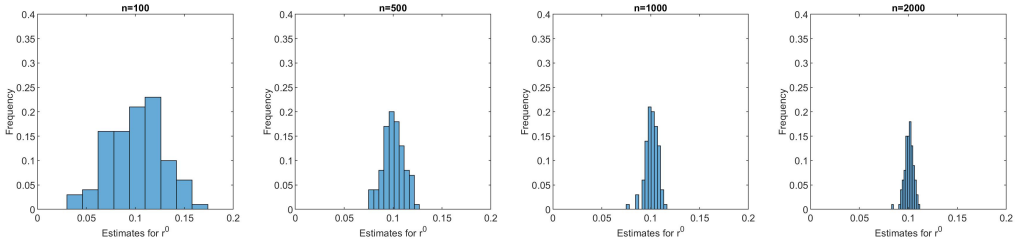

Figure 5.8: *Circle network*: histograms of the obtained estimates for r^0, with n increasing from left to right.

Experiment 5.2. *Single-node model.* As a second example, we study the single-node model as introduced in Section 5.3.1. We consider the setup with $r := r_1 = r_2$, which means that only the arrival rate is affected by the modulation, not the departure probability r. In our simulations we use the parameter values $\lambda_1 = 5, \lambda_2 = 15, r = 0.1, p_{12} = 0.1$ and $p_{21} = 0.2$. The initial values in the algorithm that maximizes the log-likelihood are based on moment estimators. The results of the maximum likelihood estimates are shown in Table 5.2 and Figure 5.9.

n	λ_1	λ_2	r
100	5.6507 (2.7007)	14.9201 (3.1332)	0.1081 (0.0259)
500	4.8028 (1.8180)	14.5595 (1.7340)	0.0991 (0.0174)
1000	5.0127 (2.3698)	14.2682 (1.9638)	0.1024 (0.0168)
2000	5.2278 (2.1442)	14.6499 (2.4113)	0.1024 (0.0150)

n	p_{12}	p_{21}
100	0.1266 (0.1525)	0.2331 (0.1869)
500	0.1240 (0.1011)	0.1977 (0.0630)
1000	0.1432 (0.1559)	0.2047 (0.1011)
2000	0.1398 (0.1629)	0.2147 (0.1136)

Table 5.2: *Single-node model*: mean of estimates of 100 data sets, with corresponding standard deviation between brackets. True parameter values: $\lambda_1 = 5, \lambda_2 = 15, r = 0.1, p_{12} = 0.1, p_{21} = 0.2$.

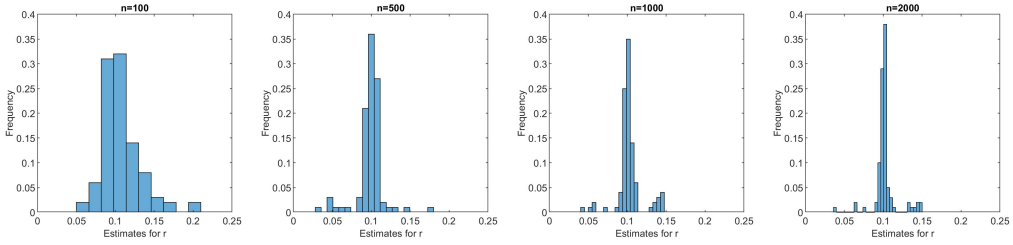

Figure 5.9: *Single-node model*: histograms of the obtained estimates for r, with n increasing from left to right.

Table 5.2 contains for each sample size (rows) and parameter (columns), the mean value of the 100 estimates, together with the corresponding standard deviation between brackets. The mean values of the estimates lie relatively close to the true parameter values, but the standard deviations fluctuate and do not always decrease in n. The histograms in Figure 5.9, featuring estimates for r, however, visually show that the estimates get increasingly concentrated around their respective averages. We observe that the values in the table are affected by outliers in the estimates. As we mentioned earlier, when maximizing the likelihood we cannot exclude the possibility of ending up in local optima. In the circle network we have not come across this phenomenon, but in our experiments

with modulation there have been a few runs in which we have. The histograms in Figure 5.9 show these outliers near 0.05 and 0.15. In the histograms of the other parameters (not included in this chapter), similar outliers appear.

To control this issue, it is advised to run the maximization algorithm for multiple different initial values of the parameters, and choose the parameter estimates that result in the highest likelihood value. Results of the maximum likelihood estimates based on this procedure, using four different, randomly chosen, initial values of the parameters, are shown in Table 5.3. Table 5.3 shows that the standard deviations improved considerably in comparison with the results in Table 5.2. In particular, the outliers have disappeared resulting in standard deviations that decrease in n.

A subtlety is that the accuracy of the saddlepoint approximation for background state i degrades when λ_i approaches 0. This is because in the regime of this arrival rate being 0, $m_k > m_{k-1}$ cannot happen, thus effectively creating a boundary state; cf. the discussion in Section 5.4.2. We followed the pragmatic remedy of imposing an explicit lower bound on the arrival rates (in our experiments we took 0.01).

n	λ_1	λ_2	r
100	5.1469 (1.9209)	15.2419 (2.4189)	0.1015 (0.0221)
500	5.0155 (0.4463)	14.9568 (0.6144)	0.1004 (0.0051)
1000	5.0690 (0.3602)	15.0828 (0.4532)	0.1011 (0.0044)
2000	5.0195 (0.2274)	15.0482 (0.3116)	0.1004 (0.0027)

n	p_{12}	p_{21}
100	0.1107 (0.0543)	0.2445 (0.1492)
500	0.1007 (0.0179)	0.2019 (0.0357)
1000	0.0987 (0.0138)	0.1971 (0.0289)
2000	0.1004 (0.0106)	0.2067 (0.0242)

Table 5.3: *Single-node model*: mean of estimates of 100 data sets, with corresponding standard deviation between brackets. True parameter values: $\lambda_1 = 5, \lambda_2 = 15, r = 0.1, p_{12} = 0.1, p_{21} = 0.2$.

Experiment 5.3. *Tandem network.* We continue by considering a two-node tandem network with modulation, as introduced in Section 5.3.2. In this experiment we assume P is known and given by

$$P = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}.$$

We run simulations for this model with true parameters $\lambda_1 = 1, \lambda_2 = 4, r^{(1,2)} = 0.1$ and $r^{(2,0)} = 0.25$. In our likelihood maximization routine we choose the initial values $\lambda_1 = 1, \lambda_2 = 1, r^{(1,2)} = 0.5$, and $r^{(2,0)} = 0.5$. Experiments with other initial values provide similar output. The results are presented in Table 5.4, showing for each sample size (rows) and parameter (columns), the mean value of the 100 estimates, together with the

corresponding standard deviation between brackets. In line with the first two experiments, we observe that the mean values in Table 5.4 lie close to the true parameter values. The standard deviation fluctuates somewhat, but this effect can again be mitigated by working with multiple initial values.

n	λ_1	λ_2	$r^{(1,2)}$	$r^{(2,0)}$
100	1.6896 (0.8324)	3.5884 (1.0424)	0.1169 (0.0157)	0.2893 (0.0415)
500	1.3161 (0.4669)	3.9394 (0.7301)	0.1076 (0.0096)	0.2692 (0.0230)
1000	1.1441 (0.4291)	3.9505 (0.5430)	0.1054 (0.0083)	0.2633 (0.0224)
2000	1.1610 (0.5968)	3.7440 (0.7026)	0.1052 (0.0103)	0.2645 (0.0258)

Table 5.4: *Tandem network*: mean of estimates of 100 data sets, with corresponding standard deviation between brackets. True parameter values: $\lambda_1 = 1, \lambda_2 = 4, r^{(1,2)} = 0.1, r^{(2,0)} = 0.25$.

5.6 Discussion and Concluding Remarks

In this chapter we considered a discrete-time multivariate population process under Markov modulation. We showed how the likelihood can be evaluated using saddlepoint approximations, and how this can be used to estimate the model parameters. We emphasize the model's high degree of generality, covering a wide variety of networks with different sizes and structures, and on top of that the possibility to include modulation. Moreover, the maximum-likelihood estimation approach is capable of estimating parameters based on observations of the network population vector only. In other words, the number of arrivals, jumps, and departures are not observed, but only the net effect of these processes together, while the modulating background process is not observed at all.

We illustrated the accuracy of the saddlepoint approximation through two examples, namely a single-node model and a tandem network. For these examples the likelihood can still be computed explicitly, and hence the explicit computation can be compared with the saddlepoint approximation. In a series of numerical tests we found that the differences between the two are typically small.

Then we investigated the accuracy of the maximum-likelihood estimation method through a number of numerical studies. We focused on three different settings corresponding to networks with different sizes and structures. In all examples accurate estimates are obtained. Moreover, working with multiple initial values to eliminate the outliers, results in standard deviations that decrease as the sample size grows.

The estimation method in general produces accurate estimates, but in a few cases the maximization ends up in a local maximum, as a consequence of the specific shape of the likelihood surface. One effective way to control it is by using multiple different initial values, choosing the outcome that results in the highest likelihood value. To be sure that the estimation method correctly tracks down modulation, it is important that the effect of

the background state is visible in the data. More concretely, one can imagine a parameter setting in for example the single-node model with two states, in which the effect of the higher arrival rate on the population size is essentially cancelled out by a higher departure probability, such that the states cannot be distinguished.

We believe that the results presented in this chapter offer various interesting opportunities for further research. In the first place, note that in our setup we assumed that the number of states d is known. Choosing d from the data is a model selection problem and falls outside the scope of this chapter, but would be worth studying in greater detail. Second, we focused on a discrete-time setting, allowing the computation of the cgfs, and thus facilitating the application of the saddle-point technique, but one wonders whether a similar approach could be followed for our model's continuous-time counterpart. The major complication is that if the background process evolves continuously in time, it is not directly clear how to compute the cgfs.

Various adaptations of our model could be considered as well. In this chapter, we considered only one type of individual, and (conditionally on a realization of the background process) all individuals move independently of each other through the network. Instead one could study multi-type models, or models with routing and departure probabilities that depend on the population vector before and/or after the transition, besides the state of the background process.

6. Concluding Remarks

In this thesis we considered various types of population processes of which the parameters are affected by an underlying background process. In the first chapter, we introduced the challenges that arise for statistical inference of the model parameters for this kind of models. In Chapters 2–5 we showed a collection of techniques to overcome these challenges, where each chapter focuses on a different model and a suitable estimation technique. We showed how the EM algorithm can be used to estimate the parameters of population processes under Markov-modulation. We used the Erlangization technique to evaluate the likelihood function for quasi birth-death processes, and applied this to mRNA data. Lastly, we introduced the saddlepoint technique and how it can be used to evaluate the likelihood function for multivariate population processes under modulation. This final chapter takes a closer look at the differences between the various models and techniques considered in the previous chapters. We obtain a clear view on the applicability and limitations of the approaches, and identify topics for follow-up research.

Models

Looking back at the various models that we analyzed throughout this thesis, we observe that they can be subdivided in two classes of models. On the one hand, we have the class of univariate, continuous-time populations processes, and on the other hand, the class of multivariate, discrete-time population processes. In the first part of this thesis, we considered various univariate, continuous-time population processes, all of which are actually within the class of quasi birth-death processes. We have seen that both the Markov-modulated population processes from Chapter 2, and the on/off-seq-L processes from Chapter 4 are special cases of a quasi birth-death process. In the last part of this thesis, that is Chapter 5, we have seen the broad class of multivariate, discrete-time population processes, in which the population processes are defined on an underlying network.

It goes without saying that one would like to develop inference techniques that are applicable to a broad class of models. At the same time, there is a natural trade-off between the generality of the model and the possibilities in terms of inference techniques. When narrowing down the class of models considered, typically one can come up with more powerful inference techniques.

Inference techniques

In this section, we reflect on the inference techniques developed in this thesis to identify their strong and weak points. In Chapter 2 we applied the EM algorithm. In situations with missing data, this is a technique that iteratively computes maximum likelihood estimates and the corresponding likelihood value. The EM algorithm is an attractive technique because of the fact that it yields accurate results, but application of the EM algorithm can be quite involved. This leads us to two main complications of the EM algorithm.

- Evaluation of the expectation- and maximization steps can become complicated or even infeasible for more general models. This became apparent in Chapter 2, Section 2.5. Here the goal was to estimate the death rate along with the other model parameters using the EM algorithm. This is complicated by the fact that the population size is only known at the observation times, and not in between two consecutive observations, while the total death rate is proportional to the population size. We were able to work around this complication by including a model assumption, namely, that the birth and death of each individual cannot occur in the same observation interval. One can imagine, however, that such a solution may not always be reasonable or solve the problem.

- For larger models, the amount of missing data increases substantially and rendering the computations will become extremely complex. This in particular holds for models like the multivariate population processes that we have studied in Chapter 5. In these models, the underlying network structure could give rise to relatively many unobserved movements of the individuals. Although it is potentially possible to apply the EM algorithm for these models, it would again involve a sizeable amount of missing data, and therefore computations will become extremely complicated.

We conclude that, while the EM algorithm may be a powerful inference technique for the class of univariate Markov-modulated population processes, it does not seem to be a suitable estimation technique for the more general class of quasi birth-death processes, where the rates can depend on the population size in various ways. Furthermore, it does not seem to be a suitable estimation technique for the class of multivariate population processes either, because of the increasing amount of missing data.

Besides the EM algorithm, we have developed two other likelihood-based techniques in this thesis, one that exploits the Erlangization technique and another that makes use of saddlepoint approximations. Unlike what is done with the EM algorithm, with these techniques we first approximate the likelihood and then maximize it numerically to find maximum likelihood estimates. The Erlangization technique is applicable to a broad class of models. We have seen that to apply this technique, one needs to compute the transient probabilities at exponential epochs. This comes down to solving a system of linear equations, which can be written in a compact matrix form. An important advantage is that the rates in these matrices can take all kinds of structures. However, there is one main complication when applying the Erlangization technique.

- The matrix P_t which describes the desired transient distribution, is evaluated in its entirety, see (3.8). Its dimensions are equal to D, the size of the state space of the joint Markov process $\{M_t, X_t\}$. This means that the size of this matrix may increase rapidly as soon as the model becomes large, for example when the phase process $\{X_t\}$ is defined on a large set of states. As a result, the computational time will severely increase, since multiple matrix multiplications have to be executed to evaluate the approximation in (3.8), or the computation can even become infeasible.

We see that the Erlangization technique does not seem to be a suitable technique for the class of multivariate population processes, since for these models, the matrix P_t will become prohibitively large as the size of the underlying network grows.

The saddlepoint technique is highly suitable for evaluating likelihood functions for network models, as long as we resort to discrete-time (rather than continuous-time) models. It relies on the computation of moment generating functions, and therefore it is a convenient technique when dealing with convolutions of random variables, as in the case of network models. Moreover, an advantage of considering the population processes in discrete time is that the likelihood function can be written as a product of matrices of smaller size, see equation (5.4); more specifically, the dimensions of these matrices are equal to the size of the state space of the background process, i.e. not of that of the joint Markov process. The fact that the saddlepoint technique relies on the computation of moment generating functions can also be a drawback, leading us to a main complication of this technique.

- It is not always obvious how to compute the necessary moment generating functions, especially if we would be interested in continuous-time processes under modulation. In this kind of models, the model parameters can switch in between two consecutive observations, instead of on the observation times only. Therefore, it may not be clear how the separate components in the convolution (5.6) are distributed, and hence, how to compute the moment generating functions.

Despite the fact that the saddlepoint technique is an effective tool for statistical inference of multivariate population processes in discrete time, we conclude that this technique does not seem suitable for the continuous-time quasi birth-death processes from Chapters 2–4.

Further research

The above discussion with respect to the various models and techniques in this thesis suggests interesting directions for further research. Naturally, the combination of the two types of models that we considered, leads to a class of models that we did not study. Namely, multivariate population processes under modulation, considered in continuous time. A question is whether it is possible to perform statistical inference for this class of models, and whether it is possible to do that using the techniques considered in this thesis. Inverse problems as studied in this thesis seem highly challenging for continuous-time multivariate population processes.

The population processes that we have studied can be extended in many other interesting ways. One may think of multi-type population processes, in which multiple

types of individuals are considered, which in turn can interact with each other. Other opportunities concern the independence assumption on the individuals in the population. Throughout this thesis, the lifetimes of the individuals, and the movement of the individuals in the multivariate processes, are assumed to be independent. However, it could also be relevant to consider models with certain dependency structures between the individuals. We see possibilities in solving inverse problems for this kind of extended models with use of the techniques considered in this thesis. Especially the saddlepoint technique seems to give possibilities for more general classes of multivariate population processes.

References

[1] Y. Aït-Sahalia and J. Yu. Saddlepoint approximations for continuous-time markov processes. *Journal of Econometrics*, 134:507–551, 2006.

[2] A. Al-Mohy and N. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM Journal on Matrix Analysis and its Applications*, 31:970–989, 2009.

[3] L. Allen. *An Introduction to Stochastic Processes with Applications to Biology.* Prentice-Hall, Upper Saddle River, NJ, USA, 2003.

[4] D. Anderson, J. Blom, M. Mandjes, H. Thorsdottir, and K. de Turck. A functional central limit theorem for a markov-modulated infinite-server queue. *Methodology and Computing in Applied Probability*, 18:153–168, 2016.

[5] D. Anderson and T. Kurtz. *Stochastic Analysis of Biochemical Systems*, volume 1.2 of *Stochastics in Biological Systems.* Springer International Publishing, 2015.

[6] H. Andersson and T. Britton. Stochastic epidemic models and their statistical analysis. *Lecture Notes in Statistics*, 151, 2000.

[7] S. Asmussen, F. Avram, and M. Usabel. The erlang approximation of finite time ruin probabilities. *ASTIN Bulletin*, 32:267–281, 2002.

[8] S. Asmussen, O. Nerman, and M. Olsson. Fitting phase-type distributions via the em algorithm. *Scandinavian Journal of Statistics*, 23(4):419–441, 1996.

[9] K. Atkinson. *An Introduction to Numerical Analysis.* 2nd edition. Wiley, Chichester, UK, 1989.

[10] O. Barndorff-Nielsen and D. Cox. Edgeworth and saddlepoint approximations with statistical applications (with discussion). *Journal of the Royal Statistical Society, Series B*, 41:279–312, 1979.

[11] N. Bingham and S. Pitts. Non-parametric estimation for the m/g/∞ queue. *Annals of the Institute of Statistical Mathematics*, 51:71–97, 1999.

[12] J. Blom, K. de Turck, and M. Mandjes. Functional central limit theorems for markov-modulated infinite-server systems. *Mathematical Methods of Operations Research*, 83:351–372, 2016.

[13] J. Blom, K. de Turck, and M. Mandjes. Refined large deviations asymptotics for markov-modulated infinite-server systems. *European Journal of Operational Research*, 259:1036–1044, 2017.

[14] J. Blom, O. Kella, M. Mandjes, and H. Thorsdottir. Markov-modulated infinite-server queues with general service times. *Queueing systems*, 76:403–424, 2013.

[15] L. Breuer and A. Kume. An em algorithm for markovian arrival processes observed at discrete times. *International GI/ITG Conference on Measurement, Modeling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, pages 242–258, 2010.

[16] L. Bright and P. Taylor. Calculating the equilibrium distribution in level dependent quasi-birth-and-death processes. *Stochastic Models*, 11:497–526, 1995.

[17] R. Butler. *Saddlepoint Approximations with Applications*, volume 22. Cambridge University Press, 2007.

[18] R. Chen and O. Hyrien. Quasi-and pseudo-maximum likelihood estimators for discretely observed continuous-time markov branching processes. *Journal of Statistical Planning and Inference*, 141:2209–2227, 2011.

[19] F. Crawford, V. Minin, and M. Suchard. Estimation for general birth-death processes. *Journal of the American Statistical Association*, 109:730–747, 2014.

[20] F. Crawford and M. Suchard. Transition probabilities for general birth-death processes with applications in ecology, genetics, and evolution. *Journal of Mathematical Biology*, 65:553–580, 2012.

[21] D. Daley and J. Gani. *Epidemic Modelling: an Introduction*, volume 15 of *Cambridge Studies in Mathematical Biology*. Cambridge University Press, 1999.

[22] H. Daniels. Saddlepoint approximations in statistics. *Annals of Mathematical Statistics*, 25:631–650, 1954.

[23] H. Daniels. The saddlepoint approximation for a general birth process. *Journal of Applied Probability*, 19:20–28, 1982.

[24] A. Davison, S. Hautphenne, and A. Kraus. Parameter estimation for discretely observed linear birth-and-death processes. *Biometrics [published online ahead of print]*, 2020, `https://doi.org/10.1111/biom.13282`.

[25] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[26] R. Eisinga, M. Grotenhuis, and B. Pelzer. Saddlepoint approximations for the sum of independent non-identically distributed binomial random variables. *Statistica Neerlandica*, 67:190–201, 2013.

[27] Y. Ephraim and W. Roberts. An em algorithm for markov modulated markov processes. *IEEE Transactions on Signal Processing*, 57(2):463–470, 2009.

[28] W. Grassmann. Finding transient solutions in markovian event systems through randomization. *Numerical Solution of Markov Chains*, 8:37–61, 1991.

[29] D. Gross and D. Miller. The randomization technique as a modeling tool and solution procedure for transient markov processes. *Operations Research*, 32:343–361, 1984.

[30] A. Häkkinen and A. S. Ribeiro. Characterizing rate limiting steps in transcription from rna production times in live cells. *Bioinformatics*, 32(9):1346–1352, 2016.

[31] S. Hautphenne and M. Fackrell. An em algorithm for the model fitting of markovian binary trees. *Computational Statistics & Data Analysis*, 70(20):19–34, 2014.

[32] S. Hautphenne, M. Massaro, and K. Turner. Fitting markovian binary trees using global and individual demographic data. *Theoretical Population Biology*, 128:39–50, 2019.

[33] N. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26:1179–1193, 2005.

[34] R. Horn and C. Johnson. *Matrix analysis. Second edition.* Cambridge University Press, 2013.

[35] A. Jensen. Markoff chains as an aid in the study of markoff processes. *Scandinavian Actuarial Journal*, pages 87–91, 1953.

[36] M. Kaern, T. Elston, W. Blake, and J. Collins. Stochasticity in gene expression: from theories to phenotypes. *Nature Reviews Genetics*, 6(6):451–464, 2005.

[37] M. Kandhavelu, H. Mannerström, A. Gupta, A. Häkkinen, J. Lloyd-Price, O. Yli-Harja, and A. Ribeiro. In vivo kinetics of transcription initiation of the lar promoter in escherichia coli. evidence for a sequential mechanism with two rate-limiting steps. *BMC systems biology*, 5(149), 2011.

[38] S. Karlin and H. Taylor. *A First Course in Stochastic Processes.* Academic Press, New York, USA., 1975.

[39] E. S. Key. Limiting distributions and regeneration times for multitype branching processes with immigration in a random environment. *The Annals of Probability*, 15:344–353, 1987.

[40] L. Kleinrock. *Queueing Systems, Volume 1: Theory.* Wiley, Chichester, UK, 1975.

[41] E. Kolaczyk. *Statistical Analysis of Network Data: Methods and Models.* Springer Series in Statistics, Springer, 2009.

[42] V. Kulkarni. *Modeling and Analysis of Stochastic Systems, 1st edition.* Chapman & Hall, London, UK, 1995.

[43] M. Mandjes and P. Taylor. The running maximum of a level-dependent quasi birth-death process. *Probability in the Engineering and Informational Sciences*, 30:212–223, 2016.

[44] W. McClure. Mechanism and control of transcription initiation in prokaryotes. *Annual Review of Biochemistry*, 54:171–204, 1985.

[45] B. Melamed and M. Yadin. Randomization procedures in the computation of cumulative-time distributions over discrete state markov processes. *Operations Research*, 32:926–944, 1984.

[46] C. Moler and C. V. Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45:3–49, 2003.

[47] M. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach.* Johns Hopkins University Press, Baltimore, 1981.

[48] C. O'Cinneide and P. Purdue. The m/m/∞ queue in random environment. *Journal of Applied Probability*, 23(1):175–184, 1986.

[49] H. Okamura and T. Dohi. Faster maximum likelihood estimation algorithms for markovian arrival processes. *Sixth International Conference on the Quantitative Evaluation of Systems, Budapest*, pages 73–82, 2009.

[50] H. Okamura, T. Dohi, and K. Trivedi. Markovian arrival process parameter estimation with group data. *IEEE/ACM Transactions on Networking*, 17:1326–1339, 2009.

[51] S. Oliveira, A. Häkkinen, J. Lloyd-Price, H. Tran, V. Kandavalli, and A. S. Ribeiro. Temperature-dependent model of multi-step transcription initiation in escherichia coli based on live single-cell measurements. *PLOS Computational Biology*, 12(10):1–18, 2016.

[52] G. Pang and Y. Zhou. Two-parameter process limits for an infinite-server queue with arrival dependent service times. *Stochastic Processes and their Applications*, 127(5):1375–1416, 2017.

[53] J. Peccoud and B. Ycart. Markovian modelling of gene product synthesis. *Theoretical Population Biology*, 48(2):222–234, 1995.

[54] X. Pedeli, A. Davison, and K. Fokianos. Likelihood estimation for the inar(p) model by saddlepoint approximation. *Journal of the American Statistical Association*, 110:1229–1238, 2015.

[55] J. Pickands and R. Stine. Estimation for an m/g/∞ queue with incomplete information. *Biometrika*, 84:295–308, 1997.

[56] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[57] V. Ramaswami and P. Taylor. Some properties of the rate matrices in level dependent quasi-birth-and-death processes with a countable number of phases. *Stochastic Models*, 12:143–164, 1996.

[58] V. Ramaswami, D. Woolford, and D. Stanford. The erlangization method for markovian fluid flows. *Annals of Operations Research*, 160:215–225, 2008.

[59] A. Reibman and K. Trivedi. Numerical transient analysis of markov models. *Computers & Operations Research*, 15:19–36, 1988.

[60] N. Reid. Saddlepoint methods and statistical inference. *Statistical Science*, 3:213–227, 1988.

[61] W. Roberts, Y. Ephraim, and E. Dieguez. On rydéńs em algorithm for estimating mmpps. *IEEE Signal Processing Letters*, 13(6):373–376, 2006.

[62] A. Roitershtein. A note on multitype branching processes with immigration in a random environment. *The Annals of Probability*, 35:1573–1592, 2007.

[63] T. Rydén. An em algorithm for estimation in markov modulated poisson processes. *Computational Statistics & Data Analysis*, 21:431–447, 1996.

[64] R. Saecker, M. J. Record, and P. deHaseth. Mechanism of bacterial transcription initiation. *Journal of Molecular Biology*, 412(5):754–771, 2011.

[65] A. Schwabe, K. Rybakova, and F. Bruggeman. Transcription stochasticity of complex gene regulation models. *Biophysical journal*, 103(6):1152–1161, 2012.

[66] D. A. Stratton. *Case Studies in Ecology and Evolution. Book in progress.* University of Vermont, 2020, `http://www.uvm.edu/~dstratto/bcor102/`.

[67] S. Tavaré. The linear birth–death process: an inferential retrospective. *Advances in Applied Probability*, 50:253–269, 2018.

[68] N. van Dijk, S. van Brummelen, and R. Boucherie. Uniformization: basics, extensions and applications. *Performance Evaluation*, 118:8–32, 2018.

[69] L. Welch. Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):1–13, 2003.

[70] J. Xu, P. Guttorp, M. Kato-Maeda, and V. Minin. Likelihood-based inference for discretely observed birth-death-shift processes, with applications to evolution of mobile genetic elements. *Biometrics*, 71:1009–1021, 2015.

Summary

Population processes are stochastic processes that record the dynamics of the number of individuals in a population, and have many different applications in a broad range of areas. Population processes are often modelled as Markov processes, and have the important feature that transitions correspond either to an increase or a decrease in the population size. These two types of transitions are often referred to as *births* and *deaths*. A specific class of population processes is the class of birth-death processes, where transitions can only increase or decrease the population by one at a time. In many real-life situations the dynamics of a population is affected by exogenous, often unobservable, factors. Therefore, this thesis considers population processes of which the parameters are affected by an underlying stochastic process, referred to as the *background process*. The aim is to find reliable inference techniques to estimate the parameters, including those related to the background process, from discrete-time observations of the population size.

The statistical inference is complicated severely by the fact that a substantial part of the process is unobserved. First, the underlying background process is not observed. Second, only the population size is observed, which is the *net effect* of all the transitions in the dynamics of the population. Last, the population size is observed in discrete time, hence the transitions in between two consecutive observations are not observed. In this thesis we show a collection of techniques to overcome these complications for a variety of population processes. The aspects in which the models differ, ask for specific inference techniques.

For a certain class of Markov-modulated population processes, we show how the well-known EM algorithm can be used to estimate the model parameters. In these models, the background process is a finite, continuous-time Markov chain and the parameters of the population process switch between distinct values at the jump times of this Markov chain. An algorithm is presented that iteratively maximizes the likelihood function and at the same time updates the parameter estimates.

A generalization of the conventional birth-death process, involving a background process, is the quasi birth-death process. We use the Erlangization technique to evaluate the likelihood function for this kind of processes, which can then be maximized numerically to obtain maximum likelihood estimates. A specific model in the class of quasi birth-death processes is a birth-death process of which the births follow a hypoexponential distribution with L phases and are controlled by an on/off mechanism. We call this the on/off-seq-L process, and use it to model the dynamics of populations of mRNA molecules

in single living cells. Numerical complications related to the likelihood maximization are analyzed and solutions are presented. Based on real-life data, we illustrate the estimation method, and perform a model selection procedure on the number of phases and on the on/off mechanism.

Last, we consider a class of discrete-time multivariate population processes under Markov-modulation. In these models, the population process is defined on a network with finitely many nodes. In addition to the births and deaths that can occur at each of the nodes, the individuals follow a probabilistic route through the network. We introduce the saddlepoint technique and show how it can be used to evaluate the likelihood function based on observations of the network population vector. The likelihood function can again be maximized numerically to obtain maximum likelihood estimates. Throughout the thesis, the accuracy of the inference methods is investigated by extensive simulation studies.

Samenvatting

Populatieprocessen zijn stochastische processen die de veranderingen in het aantal individuen in een populatie beschrijven. Ze hebben veel verschillende toepassingen in een breed scala van onderzoeksgebieden. Populatieprocessen worden vaak gemodelleerd als Markov processen en hebben het belangrijke kenmerk dat veranderingen uitsluitend overeenkomen met een toename of een afname van de populatiegrootte. Deze twee soorten veranderingen worden vaak aangeduid als *geboorte* en *sterfte*. Een specifieke klasse van populatieprocessen is de klasse van geboorte-sterfte-processen, waarin de populatiegrootte met maar één individu tegelijk kan veranderen. In veel praktijksituaties worden de veranderingen in een populatie beïnvloed door exogene, vaak niet waarneembare, factoren. Daarom beschouwt dit proefschrift populatieprocessen waarvan de parameters worden beïnvloed door een onderliggend stochastisch proces, ook wel het *achtergrondproces* genoemd. Het doel is om betrouwbare technieken te vinden om, op basis van discrete-tijds observaties van de populatiegrootte, de parameters in het model te schatten, inclusief de parameters die gerelateerd zijn aan het achtergrondproces.

Het toepassen van statistische methoden wordt bemoeilijkt door het feit dat het proces slechts in beperkte mate wordt geobserveerd. Ten eerste wordt het onderliggende achtergrondproces niet geobserveerd. Ten tweede wordt alleen de populatiegrootte geobserveerd, wat het *netto-effect* is van alle veranderingen in de populatie. Ten slotte wordt de populatiegrootte geobserveerd op discrete tijdstippen, waardoor alle veranderingen tussen twee opeenvolgende observaties niet bekend zijn. In dit proefschrift bespreken we een aantal technieken waarmee voor verschillende populatieprocessen de parameters geschat kunnen worden, ondanks deze complicaties. De punten waarop de modellen verschillen, vragen hierbij om specifieke statistische methoden.

Voor een bepaalde klasse van Markov-gemoduleerde populatieprocessen laten we zien hoe het bekende EM-algoritme kan worden gebruikt om de parameters te schatten. In deze modellen is het achtergrondproces een eindige, continue-tijds Markovketen waarbij met elke toestandswisseling de parameters van het populatieproces veranderen. Er wordt een iteratief algoritme gepresenteerd dat in elke stap tegelijkertijd de likelihood maximaliseert en de parameterschattingen verbetert.

Het toevoegen van een achtergrondproces aan de gebruikelijke geboorte-sterfte-processen resulteert in de bredere klasse van quasi-geboorte-sterfte-processen. We gebruiken de Erlangisatietechniek om de likelihood voor dit soort processen te evalueren, die vervolgens numeriek kan worden gemaximaliseerd om maximum likelihood-schattingen te verkrijgen.

Een specifiek model in de klasse van quasi-geboorte-sterfte-processen is een geboorte-sterfte-proces waarbij de geboorten hypo-exponentieel verdeeld zijn met L fasen, en die worden gecontroleerd door een aan/uit-mechanisme. We noemen dit het on/off-seq-L-proces en gebruiken het als model voor populaties van mRNA-moleculen in levende cellen. Numerieke complicaties die te maken hebben met het maximaliseren van de likelihood worden geanalyseerd en oplossingen worden gepresenteerd. Op basis van echte data illustreren we de schattingsmethode en voeren we een modelselectieprocedure uit op het aantal fasen en op het aan/uit-mechanisme.

Tenslotte beschouwen we een klasse van discrete-tijds multivariate populatieprocessen onder Markov-modulatie. In deze modellen leeft de populatie op een netwerk met een eindig aantal knooppunten. Naast dat op elk punt geboorte en sterfte kan plaatsvinden, leggen de individuen een probabilistische route door het netwerk af. We introduceren de zadelpunttechniek en laten zien hoe deze kan worden gebruikt om de likelihood te evalueren op basis van observaties van de netwerkpopulatievector. De likelihood kan opnieuw numeriek gemaximaliseerd worden om maximum likelihood-schattingen te verkrijgen. We onderzoeken de nauwkeurigheid van alle in het proefschrift geïntroduceerde schattingsmethoden aan de hand van uitgebreide simulatiestudies.